PB-410
FX-720P
ポケットコンピュータ
データバンク
活用ハンドブック
CASIO®

PB-410
FX-720P
ポケットコンピュータ
データバンク
活用ハンドブック

はじめに

コンピュータというコトバは、ある人には複雑で難解な世界を想像させ、またある人には、すべての難問を解き明かしてくれる魔法の呪文の響きに聞こえるようです。それは、コンピュータを動かすためのプログラムをはじめとするもろもろの約束ごとを、必要以上に煩しいものと考えたり、また他方、コンピュータがなんでも処理してくれるという妄想を抱いたりしているからではないでしょうか。

この困惑と気負いが、実はコンピュータというすばらしい道具を遠ざけてしまう大きな要因になっているように思えます。コンピュータを使いこなすためには、このような先入観を捨てて、まず楽しくつき合ってみることが大切です。

「ポケコン」は、小型で軽量で使いやすいパーソナルコンピュータです。しかし、通常のポケコンは、データの記録やデータの呼び出しのためにプログラムを作り、プログラムに基づいた処理を行なう必要があります。したがって、だれもが手軽にポケコンを利用できるというわけではありませんでした。

カシオ データバンク機能つきポケットコンピュータは、プログラムがなくてもすぐに活用できるポケコンです。だれでも、簡単なキー操作でデータを入力することができます。しかも、コンピュータは正確かつスピーディにデータを処理してくれます。今日からは、うろ憶えの記憶をたぐりよせることなく、正確で便利な情報の貯蔵庫ができあがります。

本書は、データバンク機能の代表的な活用法についてまとめたものです。本書を参考にして、あなただけのデータバンクをつくってください。

目　次

データバンク機能つきポケットコンピュータを使い始めるために

ポケコンは小さなサイズにもかかわらず、多くの機能を持っています。一度にすべての機能をマスターしようとしても、なかなかたいへんです。

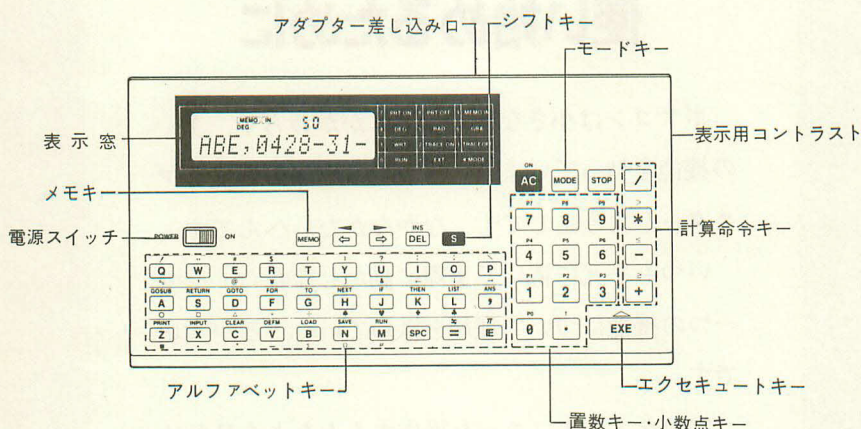
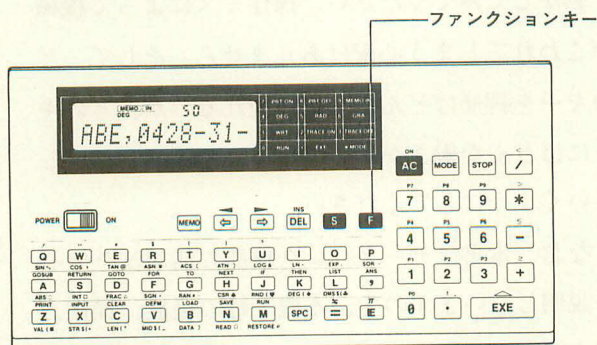
いつも手元に置いて実際に動かしながら、一つ一つの機能について理解を深めていくことが大切です。

電源を入れずにキーを操作する人をよく見かけますが、そんなことはせずに、実際に電源を入れて動かしてみてください。操作ミスによって機械がこわれてしまう心配はありません。そして、どのキーを押せばどんな表示が現れるのか、どのキーにはどんな働きがあるのかを、繰り返し体験していくようにしてください。

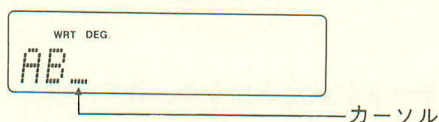
なお、本書はデータバンク機能の活用法を中心に説明していますので、本機全般の機能、操作方法また BASIC プログラムについては、別冊の**「活用ハンドブック」**をまずお読みになってください。

1 各部の名称

コンパクトに仕立てられたポケットコンピュータにも、各部それぞれ独特の名称があります。本書でもたびたび使われますので、下の図を参考にしてください。

PB-410

FX-720P

2 各部の機能

①アルファベット（大文字）の入力

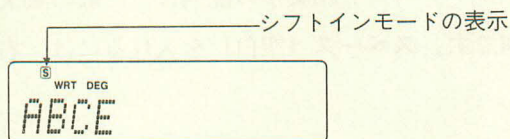
アルファベット26文字の配列は、一般の英文タイプライターに準拠した配列です。**スペース**（空白）を入れるには、右下の SPC キーを使います。

②数字の入力

数字（テンキー）の配列は、本機も電卓も同じです。もちろん本機を電卓として使うこともできます。普通の電卓とちがう点は、計算命令キーと EXE キーです。

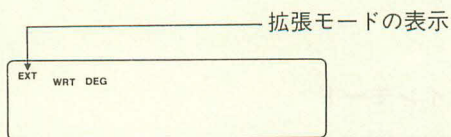
本機で掛け算を行なうときは、"×" のかわりに * （アスタリスク）を使います。割り算を行なうときは、"÷" のかわりに / （スラッシュ）を使います。また、答えを求めるときは、"＝" のかわりに EXE キーを使います。数字の "0" もアルファベットの "O" と区別するために、"0" を使います。

③シフトインモード

シフト(SHIFT)は**「切り替える」**ことを意味します。数少ないキーに数多くの機能を持たせるために、一つ一つのキーは複数の機能をにない、多いキーでは、モードを切り替えることにより、4つの機能を受け持ちます。

シフトインモードにするためには、赤い S のキー（シフトキーといい、アルファベットの S とまちがえないために、本書では SHIFT と書きます）を押します。すると、次図のようにシフトインモードのサインが表示されます。

シフトインモードが設定されると、各キーの上のパネルに赤い文字で印刷されている機能に切り替わります。たとえば**削除キー**である DEL キーは**挿入キー**である INS キーに替わります。また、プログラムでは PRINT＝表示という命令がよく使われますが、シフトインモードにすれば、P 、R 、I 、N 、T と入力するかわりに、SHIFT Z と2つのキー操作で入力することができます。

④拡張モード

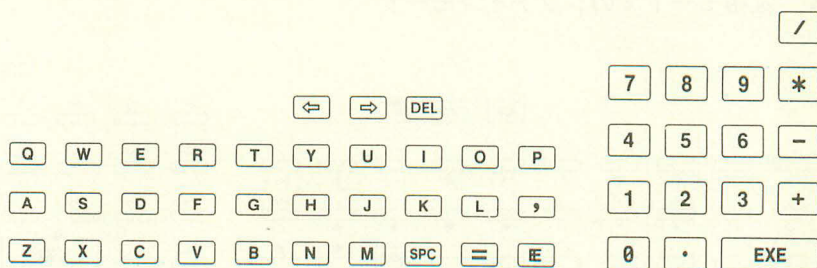
アルファベットの小文字を入力するためには、まず MODE （モード）キーを押し、続けて ・ キー（ピリオド、小数点）を押して拡張モードとします。すると、表示部に EXT（EXTENSION＝拡張）が現れます。

拡張モードで切り替わるのは、アルファベットの入力だけです。数字、DEL、カーソル移動の機能はそのままです。シフトインモードとは異なり、そのつどモードを切り替えることなく入力することができます。

⑤拡張モードでのシフトインモード

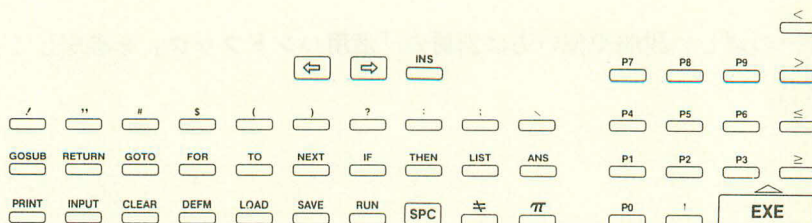
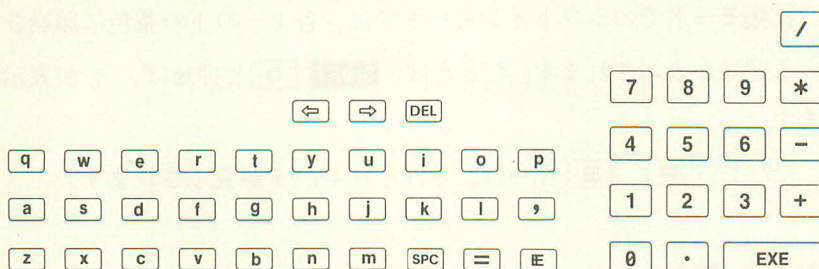
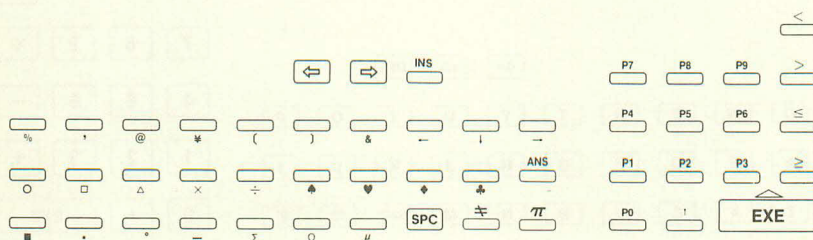
拡張モードでのシフトインモードでは、各キーの下の茶色に印刷されている記号が表示されます。たとえば、SHIFT Q と押せば、％が表示されます。

ただし、= 、E キーは、それぞれ≠、πが表示されます。

⑥モード別キー入力

1．ダイレクトモード

2．ダイレクトモードでのシフトインモード

3．拡張モード

/
7 8 9 *
⇦ ⇨ DEL
4 5 6 −
q w e r t y u i o p
a s d f g h j k l ,
1 2 3 +
z x c v b n m SPC = E
0 ・ EXE

4．拡張モードでのシフトインモード

(注) FX-720Pには、このほかファンクションモードがあります。

各キーの詳しい説明や使い方は別冊の**「活用ハンドブック」**を参照してください。

2

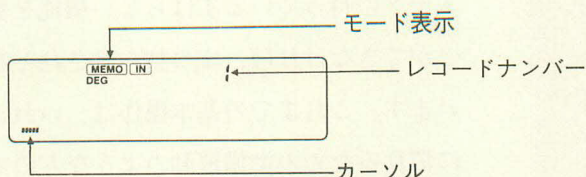
データバンク機能の使い方

前章で基本的なキー操作について説明しました。しかし、一つ一つのキー操作ができても、コンピュータの持っているすばらしい機能を働かせることができなければ、宝の持ちぐされとなってしまいます。これまでの基本操作は、いわばポケコンに慣れるための準備運動のようなものです。これからがポケコンの機能を自在に活用する実戦段階です。

すでに触れたように、データバンク機能は、プログラムの介在なしに電卓なみの手軽さでコンピュータの持っている機能を働かせることができます。もちろん、データを入力しなければコンピュータといえども働くことはできません。しかし、プログラムをつくり、そのプログラムを入力する手間をかけずにすぐ活用できるという本機の特長は、とても便利な機能です。では、さっそくパーソナル電話帳としてデータを入力し、データバンク機能の使い方を学んでみましょう。

メモインモードの指定

データバンク機能を活用するためには、まずデータを入力し、データを記憶させておかなければなりません。データの入力は**メモインモード**の指定からはじまります。

MODE キーを押し、続けて数字の 9 のキーを押せば、モードはメモインモードに替わります。表示部になにかが表示されていても、MODE 9 と押せば、メモインモードに切り替わります。表示部が下図のように変化したことを確認してください。

中央上部に MEMO IN が表示され、現在のモードがメモインモードであることを示しています。

右側の**数字の1**は**レコードナンバー**といい、メモデータの行を意味します。**カーソル**が左端で点滅し、レコードナンバー1番のデータの入力を待っている状態です。レコードナンバー1番が表示されているということは、メモデータとしては、まだなにも記憶されていないことを意味します。もし、レコードナンバー5が表示されているなら、すでに4行のメモデータが記録されていることになります。

データの入力

それでは、まず10人の名前と電話番号を入力してみましょう。

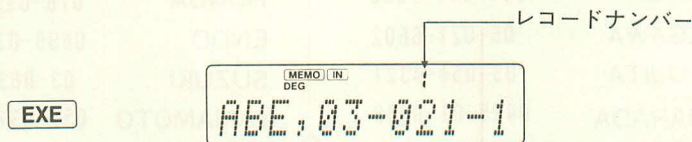
ABE	03-021-1234	HIRAYAMA	0467-01-3569
BABA	011-041-7386	HONDA	078-039-7132
EGAWA	06-021-6602	ENDO	0899-02-1007
FUJITA	03-054-4321	SUZUKI	03-063-2935
HARADA	0425-01-0038	YAMAMOTO	052-031-6221

アルファベットキーから [A] [B] [E] と名前を入力し、次に [,] キーを押します。続いて、テンキーから [0] [3] [−] [0] [2] [1] [−] [1] [2] [3] [4] と入力します。

表示部は12桁分を表示しますので、12桁を越えると、入力につれて1文字分ずつ左へずれていきます。1行目の入力が終了した段階では、下図のようになります。

操作	表示
[A] [B] [E]	MEMO IN DEG 1 ABE_
[,]	MEMO IN DEG 1 ABE,_
[0] [3] [−]	MEMO IN DEG 1 ABE,03-_
[0] [2] [1] [−] [1] [2] [3] [4]	MEMO IN DEG 1 03-021-1234_

データの入力にまちがいがあった場合は、カーソル移動キー[⇦]、[⇨]でカーソルを動かし、正しいデータを入力しなおします。

まちがいがないことを確認して、[EXE]キーを押します。これでレコードナンバー１番のデータとしてコンピュータのなかに記録されたことになります。表示部は下図のようになり、入力待ちを指示するカーソルは消えています。

もう一度[EXE]キーを押すと、表示部はクリアーされ、レコードナンバーが２番にかわり、左端にカーソルが点滅して次のデータの入力を待っている状態になります。

続いて、２行目のBABAさんの電話番号を入力します。

● [,] キーの入力

2つの例でお気づきのとおり、名前と電話番号の間には、必ず [,] キーが入っています。この [,] キーは1行のなかのデータの区切りをコンピュータに指示する命令です。後でデータを呼び出すときに重要な役目をしますので、忘れずに入力してください。

さて、ABEさん、BABAさんと同じようにして、EGAWAさんからYAMAMOTOさんまで、順次名前と電話番号を入力して「電子電話帳」に記録してください。

全部のデータの入力が終わったら、再度 [EXE] キーを押し、表示部をクリアーします。

一口メモ

連続してデータを入力する場合、[EXE] キーを二度押して表示部をそのつどクリアーする必要はありません。[EXE] キーを一回押すだけで、データは記録されます。次のデータの1文字を入力すると、表示部の前のデータは消えます。しかし [EXE] キーを押すまでは、レコードナンバーは前のままになっています。

データのサーチ

記録されているデータを読み出すことを**サーチ**といいます。サーチは英語で「**探す**」という意味です。「**サーチしなさい**」という命令を与えると、コンピュータは記録されているデータを探し出してきて表示するわけです。データバンク機能では、MEMO キーを押すことでサーチ命令をダイレクトに与えることができます。

①MEMO キーによるサーチ

MODEキーを押し、次に数字の 0 キーを押すと **RUN モード**が指定され、次のような表示になります。(電源スイッチを ON にすると、自動的に RUN モードが指定されます)

```
RUN
READY P0
```

この表示は**プログラムエリア(P)** 0番の準備ができていますというサインです。コンピュータの受け入れ体制が整って、次の命令を待っている状態を表わしています。

まず、MEMOキーを押します。表示部は下図のように変わり、レコードナンバー1番の ABE さんの名前と電話番号の一部が12文字分表示されます。

MEMO

```
RUN MEMO        1
ABE,03-021-1
```

もう一度 MEMO キーを押すと、レコードナンバー１番の電話番号だけが先頭から表示されます。

MEMO

```
RUN MEMO        1
03-021-1234
```

さらにもう一度押すと、下図のようにレコードナンバー２番のBABAさんの名前と電話番号の一部が表示されます。

MEMO

```
RUN MEMO       2
BABA,011-041
```

以上のように、MEMO キーを押すたびに、コンピュータは,で区切られた同一レコードナンバー内のデータを先頭から12文字だけ表示します。そして、同一レコードナンバー内に記録されているデータがなくなると、次のレコードナンバーに移っていきます。MEMO キーは、このように記録されているデータを,ごとに、またレコードナンバーごとに順番に表示するときに使います。

記録されているデータをすべて表示し終わると、データの入力待ちの状態になります。ここでまた MEMO キーを押すと、「**サーチしなさい**」という命令を受けた状態になり、再びレコードナンバー１番を先頭から表示します。

② EXE キーによる早送り

MEMO キーでサーチされ、表示されているデータの早送りには、EXE キーを使います。

MEMO キーを押すと、レコードナンバー1番のデータが表示されます。ここで EXE キーを押すと、レコードナンバー2番のデータが先頭から12文字分表示されます。さらに EXE キーを押すと、レコードナンバー3番のデータが表示され、先頭から12文字が表示されます（下図参照）。

EXE キーをさらに繰り返し押すと、最後のレコードナンバーを表示し、そのまま最後のデータを表示し続けます。

キー	表示
MEMO	1 ABE,03-021-1
EXE	2 BABA,011-041
EXE	3 EGAWA,06-021

このように、MEMO キーは、各レコードナンバーの全データを表示し終えてから次のレコードナンバーを表示します。一方 EXE キーは、レコードナンバーの先頭分だけを表示しただけで次のレコードナンバーに進めます。

③ SHIFT EXE キーによる逆送り

MEMO キーによっても，EXE キーによっても次のレコードナンバーに進むことはできますが、MEMO キーでは前のレコードナンバーに逆戻りすることはできません。途中までサーチした段階で、前のレコードナンバーに戻って表示させるときに、SHIFT EXE と押します。この操作により、現在表示されているレコードナンバーの前のレコードナンバーのデータを、先頭から12文字表示します。連続して行なえば、レコードナンバー1番が表示された段階で停止します。

キー操作	表示
	3 EGAWA,06-021
SHIFT EXE	2 BABA,011-041
SHIFT EXE	1 ABE,03-021-1

④条件つきサーチ

記録されている電話番号が少ないうちは、先頭のデータから順に MEMO キーによってサーチしていったり、あるいは EXE キーで一行単位に早送りしていくことも可能です。しかし、データが多くなると、これでは探し出すのに時間がかかり、不便です。

よりスピーディに、自分の必要とするデータを表示させる方法があります。条件つきサーチです。

たとえば、「E」ではじまる名前の人と電話番号を調べたい場合とか、YAMAMOTOさんの電話番号を調べたいといった場合です。

まず最初に、「E」ではじまる名前をサーチしてみましょう。

[E]と入力し、続いて [MEMO] キーを押すと、レコードナンバー3番のEGAWAさんのデータが表示されます。

続けて [MEMO] キーを押すと、レコードナンバー8番のENDOさんのデータが表示されます。

さらにもう一度 [MEMO] キーを押すと、表示部はクリアーされ、カーソルが点滅して入力待ちの状態となります。これで「E」ではじまる名前のデータのサーチは終了し、このほかには該当データがないことがわかります。

条件つきサーチの場合、該当するデータが複数あるときは、レコードナンバーの小さい順に表示されます。

同様にして、YAMAMOTOさんの電話番号を調べてみましょう。

[Y] [A] [M] [A] [M] [O] [T] [O] と入力し、[MEMO] キーを押すと、YAMAMOTOさんのデータが表示されます。

なお、条件つきサーチを行なう場合は、MEMO キーでデータを事前に表示する必要はなく、また前にサーチしたデータが表示されていてもかまいません。サーチしたい条件を入力して MEMO キーを押せば、条件に合致したデータが即座に表示されます。

このように条件つきサーチの手法で、YAMAMOTOさんのデータはすぐにサーチできましたが、電話番号は市外局番しかわかりません。条件なしのサーチの場合には、MEMO キーを押せば,で区切られた次の項目が先頭から表示されました。条件つきサーチではどうなるでしょう。

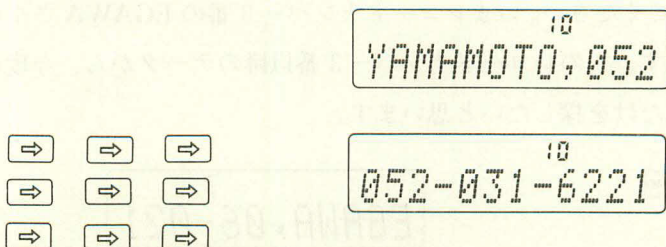
条件つきサーチの場合には、MEMO キーを押すとYAMAMOTOと同じ名前のデータがないかをサーチにいき、合致したデータがあればそれを表示し、なければ入力待ちの状態になることは、先ほどの「E」による条件つきサーチのところで述べました。それでは、YAMAMOTOさんの電話番号を表示させるには、どうしたらよいのでしょうか。

⑤カーソル移動キー ⇦ 、⇨ による確認

1つのデータが長いときは、**カーソル移動キー**を使って表示を左右に一文字ずつ移動させながら、全体を確認することができます。ただし、同一レコードナンバー内でしか作動しません。一行の先頭の文字や最後の文字まで移動させると停止します。もちろん一行のデータの長さが12桁以内のときには、この機能は働きません。

10
YAMAMOTO,052

⇨ ⇨ ⇨
⇨ ⇨ ⇨
⇨ ⇨ ⇨

10
052-031-6221

⑥ SHIFT ⇦、SHIFT ⇨ による確認

シフトインモードを指定してカーソル移動キー ⇨ を押すと , で区切られた次の項目を先頭から表示します。この機能は、MEMO キーを押したときと同じ働きをすることになります。しかし、MEMO キーの場合は、同一レコードナンバー内にデータがなければ次のレコードナンバーへ進みますが、SHIFT ⇨ の場合は、同一レコードナンバー内だけで作動し、データがなくなれば表示は停止します。

逆に、シフトインモードでカーソル移動キー ⇦ を押すと、, で区切られた前のデータを表示します。この場合も、同一レコードナンバー内だけで作動します。

```
                    10
YAMAMOTO,052
```

SHIFT ⇨

```
                    10
052-031-6221
```

⑦別条件サーチ

「E」の条件を与えて、現在 EGAWA さんのデータが表示されているとします。

このとき、続けて他の条件を与えて新たなサーチを行なうことができます。下図を見てください。いまレコードナンバー 3 番の EGAWA さんが表示されています。このレコードナンバー 3 番以降のデータから、今度は、東京在住の人だけを探したいと思います。

E MEMO

```
                    3
EGAWA,06-021
```

東京在住者は、電話の市外局番を条件として指定します。条件となる [0] [3] を入力し、[SHIFT] [MEMO] と押すと、いままでのサーチの条件であった「E」が「０３」に変更されます。そして、レコードナンバー3番の次の，以降のデータのなかから「０３」ではじまる電話番号をサーチして表示します。

表示されるのは、サーチの条件とした電話番号です。名前を表示させるには、[SHIFT] [⇦] と押します。これでレコードナンバー4番のFUJITAさんの名前が表示されます。

[SHIFT] [⇦]

4
FUJITA;03-05

さらにもう一度 [MEMO] を押せば、「０３」に該当する次のデータを探し、レコードナンバー9番のSUZUKIさんの電話番号が表示されます。

[MEMO]

9
03-063-2935

このように、別条件を与えてサーチする SHIFT MEMO 機能は、一定のルールで入力されたメモデータから、まず初期条件に該当するレコードナンバーを表示させ、次にそれ以降のメモデータに別条件を与えて選び出しを行なう機能です。この機能を上手に使うと、条件を加えながら必要なデータを絞り込んでいくことができます。

一例として、クラス別、個人別の成績ランキングのデータがあったとします。

第一項目に順位を、第二項目に個人名を、第三項目にクラス名を順番にメモデータとして登録しておきます。このとき、20位以降のA組の人の名前を調べてみたいと思います。

まず、 2 0 と数字を入力し、 MEMO キーを押して、20位の人のデータを表示します。

2 0
MEMO

20
20,MIURA,B

次に「A組」(A)と入力し、 SHIFT MEMO キーを押せば、20位以降でA組の人のデータ「A」が表示されます。

A
SHIFT MEMO

25
A

SHIFT ⇦ キーを使えば、名前を確認することができます。

SHIFT ⇦

25
SATO,A

なお、レコードナンバー順に順位に従って登録しておけば、レコードナンバーはそのまま順位を示します。

さらに続けて MEMO キーを押せば、A組の該当データが順に表示されます。そして、該当データがなくなると、カーソルが点滅して入力待ちの状態となり、サーチは完了します。

一口メモ：サーチ条件とメモデータ

条件つきサーチを行なう場合、入力されたサーチ条件とメモデータが一致していなければなりません。たとえば、メモデータがアルファベットの小文字で記録されているときに、条件を同じアルファベットの大文字で指定すると、人間にとっては同じ内容とわかりますが、コンピュータは該当データなしと判断します。

逆にこの機能をたくみに利用し、同じようなデータを項目ごとに文字をかえて入力、サーチすれば、利用度はぐんと高まります。

別条件サーチを行なうとき、SHIFT キーを押し忘れると（たとえば、単に 0 3 MEMO と押すと)、コンピュータは新しい条件によるサーチと判断して、新しい条件に該当するデータをレコードナンバー1番からサーチしなおして表示します。SHIFT キーの押し忘れに注意してください。

4 データの訂正

入力したデータのまちがいを訂正するときは、メモインモード(MODE 9)で行ないます。

入力後まだ EXE キーを押す前であれば、そのまま**カーソル移動キー**で訂正が必要な文字の下へカーソルを移動し、正しい文字に訂正してから、EXE キーを押します。これで訂正されたデータが入力されます。

EXE キーを押したあとで、たとえばサーチの実行中にまちがいに気がついたとします。前記のBABAさんの電話番号(011-041-7386)を、まちがえて011-041-7396と入力していたような場合です。

このようなときは、次の手順でデータを訂正します。

メモインモードでカーソル移動キーを押すと、図のように EDIT と表示され、カーソルとカーソルの上の文字が交互に点滅します。これは**エディット(編集)モード**になったことを示します。この状態では訂正機能が働き、データのサーチ機能は一時解除されます。

8 EXE

MEMO IN 2
BABA,011-041

EXE キーを押すことによって正しい文字が記憶され、先頭が表示されます。

こうして、データの訂正が終わると、EDIT のサインが消え、再びサーチ機能が復活します。

なお、前ページの図で、メモインモードの指定の後、MEMO キーや EXE キーを使って訂正するレコードナンバーを表示させましたが、もちろん、次のような条件つきサーチで表示させてもかまいません。データが多い場合は特に、この方法が便利です。

B A B A MEMO

MEMO IN 2
BABA,011-041

それでは、念のために、データの訂正が正しく行なわれたか確認してみましょう。

SHIFT ⇨

MEMO IN 2
011-041-7386

まちがいなく7396は7386に訂正されています。

5 データの消去

データの消去は、データの訂正と同じように、メモインモードを指定し、カーソル移動キー[⇦]、[⇨]のいずれかを押してエディット状態にしてから行ないます。

データの消去には、データの一部を消去する方法と、レコードナンバーの内容をすべて消去する方法とがあります。

①データの一部を消去する場合

まず、消去するデータを表示し、カーソル移動キーを押してエディット状態にします。

次に、消去する文字の位置へカーソルを移動させます。

カーソルと表示されている文字が交互に点滅するのを確認して、[DEL]キーを押し、続いて[EXE]キーを押します。

これでその一文字だけが記憶から消されます。表示部にはそのレコードナンバーのデータが先頭から表示され、[EDIT] 表示も消えます。

二文字以上のデータを消去する場合は、[DEL]キーで不要な文字すべてを消してから、最後に[EXE]キーを押します。

②レコードナンバーの内容をすべて消去する場合

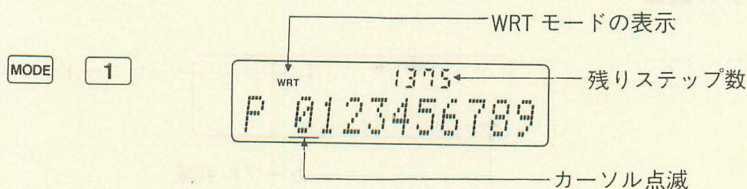
レコードナンバーの内容をすべて消去するときは、まずカーソル移動キーを押してエディット状態にします。

次に、[AC]キーを押し、続けて[EXE]キーを押します。

これでそのレコードナンバーのデータはすべて記憶から消去され、そのレコードナンバーのデータとして、次のレコードナンバーのデータが自動的に繰り上がります。

●メモデータのすべてを消去する方法

データの一部あるいは、レコードナンバーのすべてを消去する方法のほかに、メモデータのすべてを一まとめに消去する方法もあります。

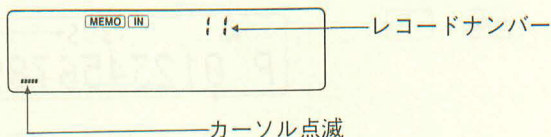
まずWRT（ライト＝書き込み）モードを指定します。WRTモードは MODE 1 で指定できます。

すると、WRTの表示があらわれ、書き込みモードになったことを教えてくれます。

1375の数字は、本機に記録できる残りのデータ容量を示しています。この数字は、データを入力するごとに減っていきます。そして、0以下になる書き込み操作をするとERR 1が表示されます。このステップ数については、別冊の**「活用ハンドブック」**を参照してください。

ここで、 N E W SHIFT # (NEW #) と入力し、EXE キーを押せば、メモインモードで入力されたすべてのデータは本機の記憶から消去されます。

データの追加

すでに記録されているメモデータに新しいデータを追加するには、記録されているデータの次に追加する場合と、記録されているデータの行間に挿入する場合とが考えられます。

①データの追加

メモインモードを指定すると、入力待ちの状態となってレコードナンバーが表示されます。上図の場合11番のレコードナンバーが表示されてカーソルが点滅し、11行目のデータの入力を待っています。これは、すでに10行のデータが記録されていることを意味しています。

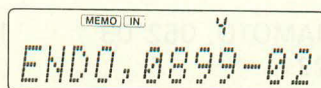
新たなデータを入力して EXE キーを押せば、レコードナンバー11番にデータが追加されます。

②データの挿入

先ほど入力した10人分の電話番号データでは、名前がアルファベット順に並んではいませんでした。このデータをアルファベット順に並びかえるためには、まずEGAWAさんとFUJITAさんの間にENDOさんのデータ行を挿入し、現在8番のレコードナンバーに記録されているENDOさんのデータ行を消去すればよいことになります。

まず、メモインモードを指定し、次に [MEMO] キーでレコードナンバー4番の FUJITA さんのデータを表示させます。本機では、挿入は表示されているレコードナンバーの前に挿入するという方法で行ないます。つまり、「この前に挿入せよ」という命令を与えるわけです。

MEMO IN
FUJITA,03-05

次に、ENDO, 0899-02-1007を入力します。

ここで、[EXE] キーを押すと、入力されたデータは記録されているメモデータの最後の行の次の行、この場合ですとレコードナンバー11番に追加されてしまいます。

データ行を挿入する場合は、[SHIFT] キーを押してシフトインモードにしてから [EXE] キーを押します。

[SHIFT] [EXE]

MEMO IN
ENDO,0899-02

これでレコードナンバー4番に ENDO さんのデータが入力されました。そして FUJITA さん以下のデータは、レコードナンバーが1つずつ繰り下がりました。[MEMO] キーを押して、FUJITA さんのデータがラインナンバー5番に表示されるのを確認してください。

次に EXE キーを押して、HONDAさんの次のレコードナンバー8番にあったENDOさんのデータを表示します。すでに1行挿入されているため、ENDOさんのデータはレコードナンバー9番で表示されます。

ここでカーソル移動キーを押してエディット状態にし、AC EXE と押せば、レコードナンバー9番のデータは消去され並びかえは完了です。

以下に挿入前のデータと挿入後のデータの出力例を掲げておきます。

挿入前	挿入後
LIST #	LIST #
1 ABE, 03-021-1234	1 ABE, 03-021-1234
2 BABA, 011-041-73 86	2 BABA, 011-041-73 86
3 EGAWA, 06-021-66 02	3 EGAWA, 06-021-66 02
4 FUJITA, 03-054-4 321	4 ENDO, 0899-02-10 07
5 HARADA, 0425-01- 0038	5 FUJITA, 03-054-4 321
6 HIRAYAMA, 0467-0 1-3569	6 HARADA, 0425-01- 0038
7 HONDA, 078-039-7 132	7 HIRAYAMA, 0467-0 1-3569
8 ENDO, 0899-02-10 07	8 HONDA, 078-039-7 132
9 SUZUKI, 03-063-2 935	9 SUZUKI, 03-063-2 935
10 YAMAMOTO, 052-03 1-6221	10 YAMAMOTO, 052-03 1-6221

7 パーソナル時刻表

以上がデータバンク機能を利用する場合の基本的な操作例です。手近な例として電子電話帳をとりあげましたが、次にもう一例、時刻表の例をあげておきます。出力例を参考にして、自分の役に立つパーソナル時刻表をつくってみてください。

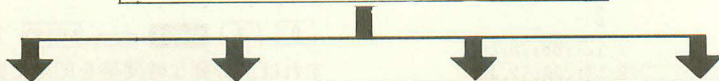
```
LIST#
    1 06:,07,25,48
    2 07:,07,26,35,48
      ,E55
    3 08:,05,08,E15,2
      7,E34,45,56
    4 09:,03,08,12,E1
      8,25,37,E59
    5 10:,05,14,28,E3
      5,48
    6 11:,E02,14,30,5
      2
    7 12:,08,26,E45
    8 13:,00,32,48
    9 14:,10,28,41
   10 15:,04,32,57
   11 16:,07,29,54
   12 17:,02,E15,20,E
      28,36,50,E58
   13 18:,03,E10,22,E
      30,45,56
   14 19:,04,E15,28,4
      4,56
   15 20:,11,E25,38,5
      4
   16 21:,15,39,59
   17 22:,20,48
   18 23:,14,36
```

時刻表は各時間帯ごとに発車時刻を表示させ、分のデータと区別するために時間の後に：(コロン)を入れておきます。この区別をつけないで、たとえば [0] [7] と条件を入れてメモデータをサーチすると、表示される07が、7時なのか7分なのか、また6時台の7分なのか、7時台の7分なのか、わかりにくくなります。

[0] [7] [SHIFT] [:] [MEMO] でサーチすれば、必要な時間帯を即座に表示します。

なお、リスト中のEは急行(EXPRESS)を表わしています。必要な時間帯を条件つきサーチした後で、Eによる別条件サーチをすれば、その時間帯以降の急行の発車時刻をすぐ知ることができます。

なお、オプションのプリンタがあれば、メモインモードで入力したデータを上のようにプリンタ用紙に印字して利用することができます。

8 データバンク機能のまとめ

ステップ数のかぞえ方

データバンク機能を使い、メモデータを記憶させていきますと、プログラムエリアとしても使っているステップ数が減っていきます。
この残りステップ数に応じたメモデータが記憶できます。
残りステップ数の確認は MODE 1 と押して、表示窓の上に点灯している数字で行なえます。
データバンク機能で使われるステップ数は1文字1ステップとなります。

例) ＡＢＥ, 03―021―1234→15文字 (，も1文字と数えます)

このほかに、一行の区切りとして EXE キーを押したときにも1ステップ必要となります。
上記の例では15文字＋ EXE として、16ステップ必要となります。

プログラムによるデータバンク機能の使い方

「コンピュータ、ソフトがなければただの箱」とよくいわれるように、コンピュータを働かせるためには、コンピュータに仕事を指示するプログラムが必要です。ポケコンも立派なコンピュータです。本機にはデータバンク機能があり、プログラムを使わなくてもかなり便利に使えますが、プログラムといっしょに使えばさらに便利になります。

たとえば、順不同で入力したデータをアルファベット順に並びかえて出力する。特定の条件を指定して、その条件に合致するデータだけを表示する。入力したデータを一定のルールにしたがって計算し、その結果を表示する。こういったことがプログラムの威力によって可能となります。

幸いコンピュータの専門教育を受けなくとも、だれでもマスターできるプログラム言語として「BASIC」が生み出され、コンピュータは、私たちにより身近な扱いやすいものとなりました。プログラムの活用によって、ぜひ本機の利用範囲を拡げたいものです。

1 プログラムによるメモデータのサーチ

①プログラムによる「一発サーチ」

MEMO キーでのサーチは、電話番号を調べたいときには、まず名前で条件サーチし、条件に合った名前が表示されてからカーソル移動キーで表示を移動させて電話番号を調べるという方法でした。

入力されているデータが、名前と電話番号という2つの項目であれば、データの引き出しに不便は感じないはずです。しかし、データ量が増え、入力する項目が3項目、4項目ともなると、呼び出すごとに MEMO キーやカーソル移動キーで表示を動かさなければならないのは、多少面倒くさい作業といえましょう。

ここでは、名前を入力すると電話番号が即座に見やすく表示される「電子電話帳プログラム」を入力して、前章ですでに入力してあるメモデータのさらに高度な活用を図っていきたいと思います。

プログラムの入力は、一字一字正確に行なってください。一字でもまちがって入力すると、実行段階でERR(エラー)が表示されてプログラムが停止したり、まちがったデータが表示されたりします。特に、**セミコロン**(;)、**コロン**(:)、**カンマ**(,)、**ダブルクォーテーションマーク**(")は、それぞれ特有の働きがありますので、入力ミスに気をつけてください。

②プログラムの入力

プログラムを入力するときは、WRT (WRT=書く)モードを指定します。

RUN (RUN=実行する)モードで入力しようとすると、入力表示だけはできますが、EXE キーを押すと「ERR 2」が表示されます。これは「まちがった入力です」と教えています。エラーが表示されたら、AC キーを

押してエラーを解除し、表示部をクリアーします。

[MODE] [1] でWRTモードを指定します。表示部は次のようになり、数字の0が点滅していると思います。

```
WRT          1375
P 0123456789
```

これで、プログラムを入力するための準備は完了です。プログラムの構成やコマンドについては、別冊の**「活用ハンドブック」**を参照していただくことにして、さっそくプログラムを入力してみましょう。

[AC] キーを押して表示部をクリアーしてください（[AC]キーを押さずにそのまま入力することもできます）。

まず、行番号10をテンキーから [1] [0] と入力します。

続いて、[R] [E] [S] [T] [O] [R] [E] と入力し、[SHIFT] キーを押してシフトインモードにして[E]キーを押します。シフトインモードの[E]は#を入力します。表示部はプログラムリストとはちがって、文字がそのまま全部つまって表示されています。

入力にまちがいないか確認してから、[EXE]キーを押すと、プログラムリストと同じように、行番号のあとに1文字分の空白ができます。

次に、行番号20を入力します。1行の入力が終ったら、必ず表示内容とプログラムを照合してから EXE キーを押すようにしてください。

EXE キーを押すと、プログラムがコンピュータに記録されます。コンピュータにプログラムが記録されたかどうかは、表示部上のステップ数が減少することでも確認することができます。

```
 10 RESTORE#              120 J=0
 20 READ# $               130 RESTORE# D$+E$+
 30 A$= MID$(1,7)             F$+G$+H$,1,180
 40 PRINT A$;             140 READ# $,$
 50 INPUT $               150 J=1
 60 IF $=" " THEN 40      160 PRINT $
 70 L= LEN($)             170 GOTO 130
 80 FOR I=0 TO 4          180 IF J=0 THEN PRI
 90 D$(I)= MID$(1,7           NT "No Data"
    )                     190 END
100 $= MID$(8)
110 NEXT I
```

ステップ数：168

●プログラムの訂正

• EXE キーを押す前：

EXE キーを押す前であれば、まだカーソルが点滅しています。カーソル移動キーでカーソルを訂正する箇所へ動かし、正しいデータを入れてから EXE キーを押します。

• EXE キーを押した後：

EXE キーを押した後でまちがいに気づいたときは、まず L I S T と入力し、続けてまちがえた行番号を入力して EXE キーを押します。まちがった行が表示されますので、訂正し、再度 EXE キーを押します。

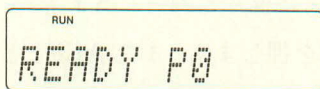
• LIST 命令を使わない方法：

訂正したい行番号と同じ行番号を入力して正しいプログラムを書き込み、EXE キーを押せば、記録されている前の行番号のプログラムは消

去され、新しく入力された内容にかわります。なお、行番号だけを入力して EXE キーを押すと、プログラムの無入力と同じことになり、その行番号にあったプログラムはすべて消去されます。

③プログラムの実行

プログラムの実行はRUNモードで行ないます。MODE 0 と操作して、RUNモードを指定します。READY P 0が表示されて、プログラムの実行が可能な状態となりました。

RUN
READY P0

プログラムを実行する命令には2種類の方法があります。

1つは、R U N あるいは SHIFT RUN と入力し EXE キーを押す方法です。

もう1つの方法は、SHIFT キーを押してシフトインモードにしてから、プログラムエリアを指定する方法です。いまは、プログラムエリアP 0にプログラムが記録されていますから、数字の 0 を押します。

どちらかの方法でプログラムを実行します。

もし、プログラムの入力にミスがあれば、ERR 2と表示され、何番の行にミスがあったかをコンピュータが教えてくれます。ERR 2は構文上のミスがあるという意味で、コマンドのスペルミスなどの指摘が多いものです。AC キーを押してエラーを解除し、WRTモードに切りかえて訂正してから、再度 MODE 0 と押してRUNモードを指定します。

さて、プログラムを実行します。プログラムは、名前を入れれば即座にその人の電話番号を表示するプログラムです。

SHIFT 0 と押します。表示部にABE？ と表示されました。

では、HARADAと入力して EXE キーを押してみます。HARADAさんの電話番号が表示されます。

今度はBABAさんの電話番号を調べてみます。

もう一度 SHIFT 0 を押します。またABE?と表示されますが、気にしないで、BABAと入力し、EXE キーを押します。

本当は、〝名前は?〟とコンピュータが聞いてくれればたいへん使いやすくなります。また、このままではABEさんの電話番号を調べることはできません。

実際に、ABEと入力してみてください。コンピュータははNo Dataと答えてきます。メモデータのなかには、確かにABEさんの電話番号は入っています。しかし、このプログラムでは、ABEさんの電話番号を読みとることができません。なぜでしょう。

原因は、行番号10番から40番にあります。このプログラムは、10番から40番で、メモデータのレコードナンバー1番の最初の項目のデータを読み

込んで、データの先頭から7文字を表示しなさいという命令を与えています。メモデータのレコードナンバー1番の最初の項目はABEさんであり、ABEは7文字以内なので、ABE?という表示になるわけです。

この解決策は簡単ですね。

メモデータのレコードナンバー1番に、〝名前〟にあたる〝NAME〟を記録しておけばよいでしょう。さっそく訂正してみます。

これで訂正は終わりです。RUN モードにして、また実行してみてください。NAME？とコンピュータが聞いてきますので、調べたい人の名前を入力して EXE キーを押します。即座に電話番号が表示されます。

```
LIST #
  1 NAME
  2 ABE, 03-021-1234
  3 BABA, 011-041-73
    86
  4 EGAWA, 06-021-66
    02
  5 ENDO, 0899-02-10
    07
  6 FUJITA, 03-054-4
    321
  7 HARADA, 0425-01-
    0038
  8 HIRAYAMA, 0467-0
    1-3569
  9 HONDA, 078-039-7
    132
 10 SUZUKI, 03-063-2
    935
 11 YAMAMOTO, 052-03
    1-6221
```

メモデータに記録されている電話番号は左表の通りです。1行目に NAME が入っています。
電話番号を調べるには SHIFT 0 でプログラムをスタートさせ、NAME？と聞いてきたら、名前を入力して EXE キーを押します。名前は頭文字の入力でもサーチできます。

④プログラムの改良

メモデータの項目を増やしたい、他の項目から名前へのサーチもしたい、こんな要望に応えてみましょう。

たとえば、会合の出欠を確認した人、まだしない人を調べるのは、なかなか面倒なものです。このチェック項目を加えてみましょう。そして、前記の電話番号を調べるプログラムを改良して、電話番号からも、また3番目のチェック項目からも名前を調べることのできる電子電話帳プログラムをつくってみましょう。もちろん、名前から電話番号を調べる機能もついています。

●メモデータの項目の追加

同じメモデータからの読み出しを行ないます。メモデータに3番目の項目を追加してデータを入力します。

	1	2	3
1	名前	電話番号	備考
2	阿部	03－021－1234	済
3	馬場	011－041－7386	未
4	江川	06－021－6602	未
⋮			
11	山本	052－031－6221	済

電話番号呼び出しプログラムの改良です。レコードナンバー1番には項目名を入れます。一番目の項目の名前〝NAME〟でした。二番目の項目は電話番号ですので〝TEL NO〟、三番目の項目は〝COMMENT〟とし、メモインモードでデータを呼び出してデータを追加します。出欠を確認した人は「済み」の意味でOVER、確認をまだしていない人には、「まだ」の意味でYETとします。

項目と項目の間には、**カンマ(，)**を入れることを忘れないでください。

メモインモードで入力しなおされたデータのリストは、次のようになっています。

```
LIST #
   1 NAME, TEL NO, COM
     MENT
   2 ABE, 03-021-1234
     , OVER
   3 BABA, 011-041-73
     86, YET
   4 EGAWA, 06-021-66
     02, YET
   5 ENDO, 0899-02-10
     07, OVER
   6 FUJITA, 03-054-4
     321, OVER
   7 HARADA, 0425-01-
     0038, YET
   8 HIRAYAMA, 0467-0
     1-3569, OVER
   9 HONDA, 078-039-7
     132, YET
  10 SUZUKI, 03-063-2
     935, OVER
  11 YAMAMOTO, 052-03
     1-6221, OVER
```

※リストの左端はレコードナンバーです。項目とはみなされません。

●プログラムの改良

このプログラムの目的は、3つの項目からデータをサーチすることです。

(i)名前から電話番号を調べる。

(ii)電話番号から名前を調べる。

たとえば03と入力すれば東京都内の電話番号の人の名前を全部表示する。

(iii)コメントの条件から名前を調べる。

たとえばYETと入力すれば「まだ」の人の名前を全部表示する。

以上のようなサーチができるようにプログラムを改良します。前に入力したプログラムはそのまま使うようにして、新しいプログラムを追加していきます。

プログラムリスト（電子電話帳）

```
10  RESTORE#
15  FOR I=0 TO 2
20  READ# $
30  A$(I)=MID$(1,7
    )
31  NEXT I
32  INPUT "Which It
    em", $
33  IF $="" THEN 32
34  N=INT VAL($)
35  IF N<1 THEN 32
36  IF N>3 THEN 32
37  IF N≠1 THEN 200
40  PRINT A$;
50  INPUT $
60  IF $="" THEN 40
70  L= LEN($)
80  FOR I=0 TO 4
90  D$(I)= MID$(1,7
    )
100 $= MID$(8)
110 NEXT I
120 J=0
130 RESTORE# D$+E$+
    F$+G$+H$,1,180
140 READ# $
145 IF D$+E$+F$+G$+
    H$≠ MID$(1,L) T
    HEN READ# $,$:
    GOTO 130
150 J=1
155 READ# $
160 PRINT $
170 GOTO 130
180 IF J=0 THEN PRI
    NT "No Data"
190 GOTO 10
200 PRINT A$(N-1);
210 INPUT $
220 IF $="" THEN 13
    0
230 O$= MID$(1,7)
240 P$= MID(8,7)
250 L= LEN(O$)+LEN
    (P$)
260 J=0
270 RESTORE# O$+P$,
    1,420
280 READ# $
290 FOR I=0 TO 4
300 D$(I)= MID$(1,7
    )
310 $= MID$(8)
320 NEXT I
330 READ# $
340 IF N≠2 THEN 360
350 IF O$+P$≠MID$(
    1,L) THEN READ#
    $: GOTO 270
360 READ# $
370 IF N≠3 THEN 390
380 IF O$+P$≠MID$(
    1,L) THEN 270
390 J=1
400 PRINT D$+E$+F$+
    G$+H$
410 GOTO 270
420 IF J=0 THEN PRI
    NT "No Data"
430 GOTO 10
```

ステップ数：552

●プログラムの説明

行番号10から行番号180までは、すでに入力したプログラムを流用しています。

行番号15は、このプログラムでは1項目から3項目までの3つの項目をサーチしますので、項目番号分として0、1、2の3つを用意しているということです。行番号32で「Which Item？」と、どの項目からサーチするかを質問してきます。

行番号31は、行番号15に対応してFOR ～ NEXT文をつくっています。行番号20と30を繰り返して読みなさいという命令です。すなわち、この場合は、NAME, TEL NO, COMMENTの3項目を読みなさいということです。

行番号32では、Which Item？と質問し、項目番号の入力を待ちます。

行番号33では、行番号32でなにも入力しないで EXE キーが押されたら、もう一度、行番号33の質問を繰り返すようにしています。

行番号34では、行番号32で入力された項目番号を数値化し、その数値を変数Nに代入します。

行番号35、36は、入力された項目番号が1より小さいとき、また3より大きいときには、行番号32へ戻ってもう一度項目番号の入力を待ちなさいという命令です。このプログラムでは、項目は1から3までしかありませんので、それ以外の項目を指定するとここでチェックされるわけです。

行番号37は、入力された数値が1以外のときには行番号200へ飛び、数値が2なら「TEL NO？」、数値が3なら「COMMENT？」と表示し、行番号210でサーチする条件の入力を待って実行するということです。

行番号40は、入力された数値が1なら、「NAME？」と表示し、入力された条件と合致する名前をサーチして、その電話番号を表示します。

●プログラムの実行

RUN モードを指定してプログラムを実行させると、Which Item ?と質問してきます。

(i)名前から電話番号を調べるとき

該当データがない場合は、"No Data" と表示します。

これで名前からの電話番号のサーチは終了します。EXE キーを押せば、最初の質問にもどって、次のサーチを行なうことができます。

(ii)電話番号から名前を調べるとき

続けて EXE キーを押せば、局番が 03の人の名前が順次表示され、該当データがなくなると「Which Item ?」の質問に戻ります。

もちろん、電話番号をすべて入力すれば、特定の該当者だけをサーチします。

(iii)コメント項目から「まだ」の人の名前を調べるとき

YETと入力し、EXE キーを押せば、「まだ」に該当する人の名前が表示されます。EXE キーを押すことにより、順次該当者名を表示することができます。

⑤電子電話帳使用上の注意

●データの追加と消去

この電子電話帳は、メモインモードで記憶されたデータをプログラムで呼び込んで活用しているので、データの追加、消去はメモインモードで行なってください。

●サーチの条件

プログラムでサーチできるメモデータは、1項目あたり30文字以内にしてください。1項目30文字以上になりますと、サーチ不能となり、エラーが表示されます。

また、データのサーチ条件は、1項目は30文字すべてを照合しますが、2項目、3項目は、先頭から14文字までしか照合しません。

7 データバンク機能のための専用コマンド

コンピュータに命令を与える言葉をコマンドと呼びます。本機には通常のBASICのコマンドのほかに、データバンク機能専用のコマンドがあります。

このコマンドのおかげで、メモデータをプログラムで読んだり、処理したりすることができるわけです。この専用コマンドにも、通常のコマンドと同じように、プログラム中で使うコマンドと、プログラム中では使えないコマンドの2種類があります。

①マニュアルコマンド

● LIST# (リストクロスハッチ)

メモデータの内容を、レコードナンバーの小さい順に、表示を移動させながら連続的にすべて表示します。RUNモードとWRTモードで実行できますが、メモインモードでは実行できません。

表示を途中で停止させるときは、STOPキーを押します。表示を続けるときは、EXEキーを押せば再開します。

●メモデータのプリントアウト

メモデータを**プリントアウト**（印字）させるためには、オプションのミニキャラクタープリンタ〈FP-12S〉を利用します。

本機の電源をOFFにしてから、ミニキャラクタープリンタのコネクターに差し込みます。外すときも、電源は必ずOFFにしてください。

※ 電源スイッチをONのままプリンタに接続しますと、プログラムやデータの内容に変化が生じることがあります。

メモデータをプリントアウトするときは、まず MODE 7 と押して、PRT（プリント）モードにします。表示部に PRT と文字が表示されますから確認してください。次に、LIST # と入力し、EXE キーを押せば、メモデータがプリントアウトされます。

プリントモードの解除は、MODE 8 と操作します。

● SAVE # と LOAD #（セーブ／ロードクロスハッチ）

データバンク機能つきポケコンは、RAM カードを自由に取りかえることができます。しかし、思いがけず RAM カードが不足してしまったときや電池交換時などには、カセットテープにデータを一時的に保存し、必要に応じてまたポケコンに戻すことができます。

メモデータをカセットテープに記録させるときには SAVE # を使い、カセットテープからデータを呼び出すときには LOAD # を使います。

●カセットテープレコーダーの接続

カセットテープにセーブしたり、ロードしたりするためには、別売のカセットインタフェイス〈FA-3〉とカセットテープレコーダーが必要です。

まず本機の電源を OFF にしてから、カセットインタフェイスのアダプターに差し込み、カセットインタフェイス裏面の固定スイッチを LOCK の状態にします。

次に、カセットインタフェイスから出ているコードをカセットレコーダーに接続します。白いプラグをカセットレコーダーの MIC（マイク）端子に、黒いプラグをイヤホーン端子に、灰色のミニミニプラグを REM（REMOTE＝リモート）端子に差し込めば、テープレコーダーとの接続は完了です。

カセットレコーダーのボリュームを最大にします。トーンコントロールがついていれば中位にして録音状態にします。

●メモデータのセーブ

RUNまたはWRTモードを指定します。

SAVE #と入力し、続いてダブルクォーテーションマーク(" ")で囲んでファイル名を入力します。ファイル名は8文字以内で、他のメモデータと区別しやすい名前をつけます。たとえば、電子電話帳なら"DENWA"と入力し、テープレコーダーを録音状態にして EXE キーを押すと、テープが回転してメモデータがテープにセーブされます。

リモート機能のないテープレコーダーをお使いのときは、EXE キーを押す前に、録音状態でスタートさせます。

セーブ中は表示部の表示が消え、ステップ数表示の最後の桁の場所に「—」が表示されます。

セーブが終了するとテープが止まり、表示部にはRUN、WRTモードの最初の表示があらわれます。

※ファイル名が8文字を越えたり、" "を忘れると、ERR 2が表示されセーブはできません。

ファイル名は省略してもかまいません。

●メモデータのロード

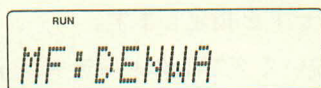
カセットレコーダーをセーブのときと同じように接続し、再生状態にします。

RUNあるいはWRTモードを指定します。

LOAD #と入力し、続いてファイル名を" "で囲んで入力し、EXE キーを押します。テープが回転し、メモデータは本機にロードされます。

リモート機能のないテープレコーダーをお使いのときは、EXE キーを押した後に、再生状態でスタートさせます。

ロード中は下記のように、表示部にメモファイル名が表示されます。

```
RUN
MF:DENWA
```

ロードが終了するとテープが止まり、表示部にはRUN、WRTモードの最初の表示があらわれます。

※ファイル名は、セーブしたときと同じファイル名を必ず入力してください。

ファイル名を省略した場合は、テープからロードされた最初のメモファイルを読み込みますので、一本のテープに数種類のメモファイルがセーブされている場合には、意図しないメモファイルが読み込まれる可能性があります。

※すでにメモデータが記憶されている場合は、自動的に一度すべてクリアーしてから新たなメモデータを読み込みます。

● NEW #（ニュークロスハッチ）

メモデータをすべて消去します。WRTモードで実行します。

NEW # EXE を実行するときには、必要なメモデータはあらかじめLIST #で印字してとっておくか、SAVE #でカセットテープにセーブしておいてください。

NEW #は、メモデータを消去して新しいメモデータを入力するときに使います。メモインモードでのデータの記入は、常に前のメモデータの次に追加されます。内容の異なるメモデータが記録されていると、サーチのときなどに不便になります。新しいメモデータを入力するときは、不要なデータを消去してからにしてください。

②**プログラムコマンド**

プログラムコマンドは、マニュアルコマンドとは異なり、それ自体で単独に使われることはなく、プログラムのなかで使われるコマンドです。本機には、通常のプログラムコマンドのほかに、特別なコマンドが用意されています。これらのコマンドをプログラム中で利用することにより、メモデータの読み込み、メモデータのサーチ、順番の設定、メモデータの書き替え、削除が可能となります。

● READ #（リードクロスハッチ）

通常のREAD命令は、プログラムのなかに書かれたDATA文を読み込むための命令ですが、READ #は、メモインモードで書き込まれたメモデータを変数に読み込む機能があります。

READ #　変数名

と指定します。変数名は、，で区切って複数個指定することができます。

READ #は、メモデータを順に変数に読み込み、次のプログラムの命令に従って実行します。しかし、メモデータが文字型で、READ #に続く変数が数値変数というようにタイプが異なる場合には、タイプミスマッチでエラー（ERR 2）となります。また、メモデータに読むべきデータがない場合にもエラー（ERR 4）となります。

しかし、READ #に続く変数に文字変数や$が使われている場合には、メモデータが数値でも文字でも読み込むことができます。

READ #で必要な数のメモデータを読み込んだ後で、再度READ #が使われると、後のREAD #で読み込まれるデータは、前のREAD #で読み込まれた次のメモデータが読み込まれます。

メモデータが1，␣␣2というスペースを含む場合は、2の前のスペースは読みとばして変数だけを読み込みます。スペースも読み込ませるためには、ダブルクォーテーションで囲って1，"␣␣2"とする必要が

あります。こうすることによって、ダブルクォーテーションで囲ってある文字列を読むことができます。

● RESTORE #（レストアクロスハッチ）

通常のRESTORE命令は、READで読むDATA文のデータを指定する命令です。一方、RESTORE #は、メモデータをサーチしたり、READ #で読み込むメモデータの順番を設定する機能があります。

RESTORE #

RESTORE #だけで次の検索文字列が省略された場合は、次のREAD #で読むメモデータの順番を、先頭から読み込むように設定します。

RESTORE #　検索文字列

RESTORE #の次に〝H〟などの検索文字列が続く場合は、〝H〟という検索文字列を先頭に含むメモデータをサーチし、サーチしたメモデータを次に実行されるREAD #のメモデータとして読み込むようにします。この場合、該当するメモデータがないとエラー(ERR 4)となります。

RESTORE #　検索文字列, 0

これは、RESTORE #　検索文字列と同じです。

RESTORE #　検索文字列, 1

検索文字列をサーチし、検索文字列を含む行の先頭から、READ #で読み込むようにします。この場合も該当するメモデータがない場合はエラー(ERR 4)となります。

RESTORE #　検索文字列,｛0または1｝,｛行番号　または
#プログラムエリア番号｝

検索文字列のサーチをし、もし該当するメモデータがない場合には、指定された行番号またはプログラムエリアにジャンプさせます。

● WRITE #（ライトクロスハッチ）

WRITE＃　メモデータ

メモデータの書き替えや削除の機能です。「WRITE #　メモデータ」で、現在 READ #で読み込もうとするメモデータを書き替えます。

A、B、Cのメモデータがあり、次の READ #でBが読み込まれる場合、「WRITE #　1，2」とすると、メモデータはA，1，2と書き替えられます。

また同じ条件で、「WRITE #　〝1，2〞」とすると、Bは1，2に書き替えられ、A，B，Cのメモデータは、A，1，2，Cとなります。

「WRITE #」とすると、現在 READ #で読み込もうとするメモデータを含む行全体を削除します。

以上が、データバンク機能で使われる特別な専用コマンドです。

7 パスワード

パスワードは、データやプログラムの機密性を保持するための機能です。キャッシュカードの暗証番号と同じような性格のものです。データバンクですから個人のプライベートな情報が記録され、他人に知られたくないこともあります。また、苦心して独自に開発したプログラムには、他人に盗用されたくないものもあります。ましてや、貴重なデータやプログラムが、他人の不注意な操作によって破壊されてはたまりません。

こうしたことを未然に防ぐために、パスワードがガードマン役を果します。

①パスワードの設定

パスワードは、WRTモード、RUNモードのどちらでも設定することができます。メモインモードでは設定できません。

まず、RUNモードを設定します。

次に、PASSと入力し、独自のパスワードを8文字以内にして、ダブルクォーテーションで囲んで入力します。

PASS "CASIO"

ここで EXE キーを押せば、パスワードが設定されます。パスワードを再度入力してパスワードを解除しないかぎり、もうメモインモードで入力したメモデータも、データを処理するプログラムもチェックすることはできません。

②パスワードの選び方

パスワードは、文字〔数字、記号、あるいはそれらの組み合わせ〕で自由につくることができます。ただし、あまり複雑なパスワードを選定してしまうと、自分自身でも忘れてしまうことになりかねません。

③パスワードの効果

パスワードが設定されると、各モードでの次の機能が働かなくなり、操作をするとエラー（ERR 8）が表示されます。

メモインモード	メモデータの書き込み
	メモデータの訂正
	メモデータのサーチ
WRT モード	プログラムの入力
	プログラムの訂正
	プログラムの追加
	LIST, LIST ALL
	LIST#
	NEW, NEW ALL, NEW#
RUN モード	メモデータのサーチ
	LIST, LIST ALL
	LIST#

以上のようにパスワードが設定されると、データバンク機能のほとんどが動かなくなってしまいますが、RUN モードでのプログラムの実行だけはできます。

④パスワードの解除

パスワードでロックされた機能を回復させるためには、パスワードを設定したときと同様に、PASS と入力し、続いてダブルクォーテーションで囲んだパスワードを入力して EXE キーを押します。パスワードが解除されると RUN、WRT モードの初期表示があらわれます。

※もし不運にもパスワードを忘れてしまったら、二度とパスワードは解除できません。このときはプログラムやメモデータをすべて消してしまう「オールリセットボタン」を押すしか方法はありません。このボタンは、本機裏面にあります。この穴を先の細い棒などで押してください。

4 スケジュール管理プログラムⅠ

電子電話帳のプログラムは、MEMOキーを使って入力されたデータをプログラムにより条件つきサーチしました。本機はまた逆に、プログラムによって入力されたデータをMEMOキーを使って簡単に見ることもできるという特長を持っています。

この機能をパーソナルスケジュール管理に応用して、一週間分のスケジュールを表示するプログラムをつくってみたいと思います。

①プログラムの入力

P0にデータの入力プログラムを、P1にデータの出力プログラムを入力してみます。

・WRTモードにして SHIFT 0 を押し、プログラムエリアP0を確保します。次頁のプログラムリストにしたがって、正確に入力してください。

行番号30の♥は、MODE ・ と押して拡張モードにしてから、SHIFT J で入力できます。アルファベットの小文字も、同様に拡張モードで入力できます。

データ入力用のプログラムの入力が終ったら、データ出力のプログラムを入力します。 SHIFT 1 を押しプログラムエリアP1を確保して、出力プログラムを入力します。

プログラムリスト

```
P0
 10 BEEP: INPUT "M
    onth/Day ", $: IF
    $="" THEN END
 20 BEEP: FOR I=0
    TO 4 : A$(I)=MID
    $(I * 7+1, 7): NEX
    T I
 30 INPUT "Schedule
    ",$ : BEEP : RES
    TORE # " ♥ ",, 40
 40 WRITE # A$+B$+C$
    +D$+E$+","+$+",
    ": GOTO 10
```

ステップ数：117

```
P1
 10 BEEP: CLEAR :
    INPUT "M/D?  Sch
    ed", $: IF $=""
    THEN END
 20 L$=MID$(1, 7):
    FOR I=1 TO 4 : S$
    =MID$(I, 1)
 30 IF S$="" THEN I
    F J≠I-1 THEN 70
 40 IF S$="/" THEN
    J=I : K=K+1 :  GOTO
    70
 50 IF S$≧"0" THEN
    IF S$≦"9" THEN
    70
 60 K=K+5
 70 NEXT I :  IF K>I
    THEN K=2
 80 IF (K-2) * (J-1) *
    (J-4) * J=0 THEN
    GOSUB 200:  GOTO
    10
100 D=VAL($):E$=M
    ID$(J+1, 2):E=V
    AL(E$)
110 L$=STR$(D)+"/"
    +STR$(E) : G=0:
    GOSUB 200: IF G
    =0 THEN BEEP :
    PRINT L$
120 F=F+1: IF F>6 T
    HEN 10
130 E=E+1: IF E<29
    THEN 190
140 IF D=2 THEN 180
150 IF E<31 THEN 19
    0
160 IF (D-4) * (D-6) *
    (D-9) * (D-11)=0
    THEN 180
170 IF E<32 THEN 190
180 D=D+1: E=1 : IF
    D>12 THEN D=1
190 GOTO 110
200 RESTORE #
210 RESTORE # L$,, 26
    0: READ # $ : M=L
    EN(L$) : M$=MID$
    (M+1, 1)
220 IF K=1 THEN IF
    M$≧"0" THEN IF
    M$≦"9" THEN 210
230 BEEP : G=1 : REST
    ORE # " ", 1
240 READ # $ :  IF $=""
    THEN STOP : G
    OTO 210
250 PRINT $+"  "; :  G
    OTO 240
260 RETURN
```

ステップ数：491

合計608ステップ

②スケジュールの入力

RUNモードに切りかえて SHIFT 0 と押し、プログラムエリアP0のプログラムをスタートさせます。

ここでは、次のスケジュールを入力してみましょう。

6月28日　誕生日
6月29日　休暇（熱海へ）
〜
7月5日
7月3日　10時江の島へ
7月4日　水曜日　野球
7月6日　東京9時：カシオ

操作		表示	
SHIFT 0		Month／Day ?	月日を入力します
6／28	EXE	Schedule ?	スケジュールを入力します
TANJOBI	EXE	Month／Day ?	
6／29-7／5	EXE	Schedule ?	
YASUMI(ATAMI)	EXE	Month／Day ?	
⋮		⋮	

以上のように、月日とスケジュールを交互に聞いてきますので、それに答えるかたちで入力していきます。

月日の入力は、月と日の間に斜線（スラッシュ）を入れておきます。

月日、スケジュールの入力はそれぞれ30文字までできます。ですから月日に続けて曜日や時間をいっしょに入力することもできます。

スケジュールの入力が終わりましたら、「Month／Day?」が表示されているときに EXE キーだけを押します。

メモデータのリスト

```
6/28,TANJOBI,

6/29-7/5,YASUMI(ATAM
I),

7/3 10:00,GO TO ENOS
HIMA,

7/4 WED,BASEBALL,

7/6,TOKYO,9:00 CASIO,
```

③プログラムの実行

まず、6月27日から一週間分のスケジュールを調べてみます。

操作		表示	
SHIFT 1		M／D? Sched?	サーチする条件を入力します。
6／27	EXE	6／27	
	EXE	6／28 TANJOBI	
	EXE	6／29-7／5 YASUMI(ATAMI)	
	EXE	6／30	
	EXE	7／1	
	EXE	7／2	
	EXE	7／3 10:00 GO TO ENOSHIMA	
	EXE	M／D? Sched?	

このように EXE キーを押すたびにその日からのスケジュールが表示されます。一週間分の表示が終了すると最初の表示の〝M／D？ Sched？″に戻ります。

●特定条件でのサーチ

"T"という特定の条件を与えてサーチすることもできます。この場合は、月日のかわりにTと入力して EXE キーを押せば、スケジュールの内でTに該当する(Tから始まる)データが先頭の日付から表示されます。

さらに EXE キーを押せば、次の該当データが表示されます。該当データがない場合、あるいはすべての該当データを表示したときには、"M／D? Sched ?" の表示に戻ります。

④データの消去

データの消去はプログラム上ではできません。データの消去はメモインモード(MODE 9)で行ないます。

データが増えてメモリーの容量を越えそうになると、エラー(ERR 1)が表示されます。このときはメモインモードで不要なデータを消去してください。消去の仕方については 24ページを参照してください。

すべてのデータを消去する場合は、WRT モードで NEW # EXE と操作ます。

RAMカードによるプログラムの発展

ポケコンのメモリー容量には制限があります。プログラムやデータをカセットテープでセーブしたりロードしたりすることもできますが、テープレコーダーやインタフェイスが必要になります。自由に持ち運んで手軽に活用するには不便が多すぎます。

その点、RAMカードですとかさばらず、どこにでも持ち運びができます。また交換はワンタッチ。ポケコン本来の携帯性、簡易性、機動性がさらに高まります。

利用目的に合わせて、RAMカードにプログラムやメモデータを記録しておけば、1台のポケコンが多方面で多種多様な働きをします。

RAMカードのライブラリーを整理し、RAMカードを上手に使い分けることができれば、本機の活用分野はますます拡がり、文字通りあなたの片腕となることでしょう。

RAMカード

RAMは、Random Access Memoryの頭文字をとったもので、任意に呼び出し、書き込みのできる装置です。

この記憶装置をカードに収納したのがRAMカードです。RAMカードは、電池でバックアップされていますので、本機から外しても記憶が消えてしまうことはありません。

したがって、ビジネスやホビーなどの利用目的に合わせてプログラムやメモデータをRAMカードに記録しておき、必要に応じて交換することができます。この交換により、本機はミニサイズにもかかわらず、多種多様なデータをスピーディに処理することができます。

RAMカードには、記憶容量が2Kバイトのカード〈RC-2〉と4Kバイトのカード〈RC-4〉の2種類があります。プログラムの大きさやデータ量に合わせて使い分けてください。

以下、2Kベース、4KベースそれぞれのRAMカードを使ったスケジュール管理のプログラムをつくってみました。あなたのRAMカードにも入力してみてください。

スケジュール管理プログラムII

前章のスケジュール管理プログラムは、日付データを入力すると指定した日から1週間分のスケジュールを表示し、またある条件を指定すると該当するデータを順次表示するプログラムでした。

ここに登場するスケジュール管理プログラムIIは、日付、時間、スケジュールを入力すると、自動的に日付・時間のスケジュール表が作成されます。データのサーチは、日付を入力すると該当日のスケジュールを時間の早いものから順に表示します。

①プログラムの入力

プログラムエリアP0には、スケジュールを記入するプログラムを、プログラムエリアP1には、スケジュールを表示するプログラムを入力します。

WRTモードでP0を指定し、次のプログラムを入力してみましょう。

```
P0
  10 INPUT "Date",$
  20 IF $="" THEN EN
     D
  30 D= VAL($)
  40 A= INT(D/100):B
     = INT(D-A*100)
  50 IF A<1 THEN 10
  60 IF A>12 THEN 10
  70 IF B<1 THEN 10
  80 IF B>31 THEN 10
  90 D=A*100+B
 100 INPUT "Time",$
 110 IF $="" THEN 10
     0
 120 T= VAL($)
 130 A= INT(T/100):B
     = INT(T-A*100)
 140 IF A<0 THEN 100
 150 IF A>23 THEN 10
     0
 160 IF B<0 THEN 100
 170 IF B>59 THEN 10
     0
 180 T=A+B/100
 190 INPUT "Schedule
     ",$
 200 RESTORE# : REST
     ORE# "\END",0,2
     10
 210 WRITE# "\END,"
 220 WRITE# "\\","\E
     P": RESTORE# :
     RESTORE# "\\"
 230 WRITE# STR$(D)+
     ","+ STR$(T)+",
     "+$
 240 RESTORE#
 250 READ# A$
 260 IF A$="\END" TH
     EN 390
 270 A= VAL(A$)
 280 READ# B,$
 290 IF A<D THEN 250
 300 IF A>D THEN 320
 310 IF B<T THEN 250
 320 RESTORE# : REST
     ORE# STR$(A)+",
     "+ STR$(B)+","
 330 WRITE#
 340 RESTORE# "\EP"
 350 WRITE# STR$(A)+
     ","+ STR$(B)+",
     "+$
 360 WRITE# "\EP"
 370 F=1
 380 GOTO 240
 390 WRITE#
 400 RESTORE# "\EP"
 410 WRITE# "\END"
 420 GOTO 10
```

ステップ数：507

②スケジュールの入力

次のスケジュールを入力します。

日　付	時　間	スケジュール
601	1000	SEIHIN　KAIGI
601	1200	BUCHO　UCHIAWASE
601	1800	EIGA
602	700	TSURI
603	915	CHOREI
604	730	CHOSHOKUKAI
605	900	SHOHIN　KAIGI
605	1500	TOKYO　EKI
607	1430	OJI　NI　DENWA

RUN モード、SHIFT 0 でプログラムをスタートさせます。

以上で入力の1サイクルは終わりです。次に再び「Date ?」と聞いてきますので、同じ方法で入力します。

入力が終わりましたら，「Date ?」と表示されているときに EXE キーだけを押します。

なお、データの入力は、同じ様式であれば、メモインモードからもできます。しかしデータの並び替えはできません。

③入力データのチェックと消去

入力データは、MEMOキーを使って確認することができます。日付、時間をランダムに入力しても、コンピュータが日付、時間の順に並び替えて記録していきます。

データの消去は、メモインモード（MEMO 9）でメモデータの消去と同じ方法で消去します。入力されたデータは、日付、時間の順に並んでいますので、MEMOキーで消去するデータを表示させ、消去します。(24ページ参照)

記録されているすべてのデータを消去するときは、メモデータの消去と同じように、WRTモードでNEW # EXE と操作します。

④スケジュール表示プログラムの入力

このプログラムは、月日を指定して、P0で入力されたその日のスケジュールを調べるプログラムです。

WRTモードで SHIFT 1 を指定し、次のプログラムを入力してみましょう。

```
P1
  10 INPUT "Date",$
  20 IF $="" THEN 10
     0
  30 D= VAL($)
  40 A= INT(D/100):B
     = INT(D-A*100)
  50 IF A<1 THEN 10
  60 IF A>12 THEN 10
  70 IF B<1 THEN 10
  80 IF B>31 THEN 10
  90 D=A*100+B
 100 K=0
 110 RESTORE#
 120 RESTORE# STR$(D
     )+",",0,260
 130 READ# A
 140 K=-1
 150 READ# B:T=B
 160 E= INT(A/100):F
     =A-E*100
 170 G= INTB:H= FRAC
     B*100
 180 A$= STR$(E)+"/"
     + STR$(F)
 190 B$= STR$(G)+":"
 200 $="0"+ STR$(H)
 210 B$=B$+ MID$( LE
     N($)-1)
 220 READ# $
 230 PRINT A$;",";B$
 240 PRINT $
 250 GOTO 120
 260 IF K=0 THEN PRI
     NT "No Data"
 270 GOTO 10
```

ステップ数：316
合計823ステップ

⑤プログラムの実行

RUN モード、SHIFT 1 でプログラムを実行します。

操 作		表 示	
SHIFT 1		Date ?	入力のときと同じように日は必ず2桁で入力します。
601	EXE	6/1, 10:00	月日と時間が表示されます。
	EXE	SEIHIN KAIGI	スケジュールが表示されます。
	EXE	6/1, 12:00	続けて EXE キーを押すと、同じ日の次のデータが表示されます。
	EXE	BUCHO UCHIAWASE	
	EXE	6/1, 18:00	
	EXE	EIGA	
	EXE	Date ?	該当データがなくなると、再び Date ? と聞いてきます。

該当データがない場合には、"No Data" と表示され，再び Date ? と聞いてきます。

⑥プログラムの改良（データの消去）

データの消去はメモインモードでしかできませんでした。ここで、RUNモードでも消去できるようにプログラムを変更します。プログラムの変更は、すでにP1に入力されている出力プログラムに改良を加えます。

WRTモードを指定し、SHIFT 1 と押します。

次のプログラムを入力します。

```
245 GOSUB 300
300 INPUT "Delete(Y
    /N)",N$
310 IF N$≠"Y" THEN
    RETURN
320 RESTORE# : REST
    ORE# STR$(D)+",
    "+ STR$(T)+","+
    $
330 WRITE#
340 RETURN
```

ステップ数：72
合計895ステップ

それでは、RUNモードにし、SHIFT 1 で実行してみましょう。

データが消去されたかどうか、メモインモードで確認してください。

⑦プログラムの改良（データの訂正と消去）

スケジュールは変更されることが多いものです。次にスケジュールの訂正もできるように、プログラムを改良してみましょう。

訂正できるのはスケジュールだけとします。月日、時間を含む訂正は、P0の入力プログラムで新しいデータを入力し、P1の表示プログラムで不要のデータを消去することとします。

WRT モードにし、SHIFT 1 を指定します。

次のプログラムを入力してください。⑥で入力したプログラムは消されます。

```
300 INPUT "D,M,[EXE
    ]",N$
310 IF N$="D" THEN
    350
320 IF N$≠"M" THEN
    RETURN
330 INPUT "Modify(Y
    /N)",M$
340 GOTO 360
350 INPUT "Delete(Y
    /N)",M$
360 IF M$≠"Y" THEN
    RETURN
370 RESTORE# : REST
    ORE# STR$(D)+",
    "+ STR$(T)+","+
    $
380 IF N$="D" THEN
    WRITE# : RETURN
390 READ# A,B
400 INPUT "Schedule
    ",$
410 WRITE# $
420 RETURN
```

ステップ数：175

合計1005ステップ

実行中、スケジュールの表示の後で EXE キーを押すと、今度は次のように表示されます。

ここで、新しいスケジュールを入力して EXE キーを押せば、スケジュールの訂正は完了します。

なお、Schedule？の質問に対してなにも入力せずに EXE キーだけを押すと、スケジュールが消去されたのと同じことになります。

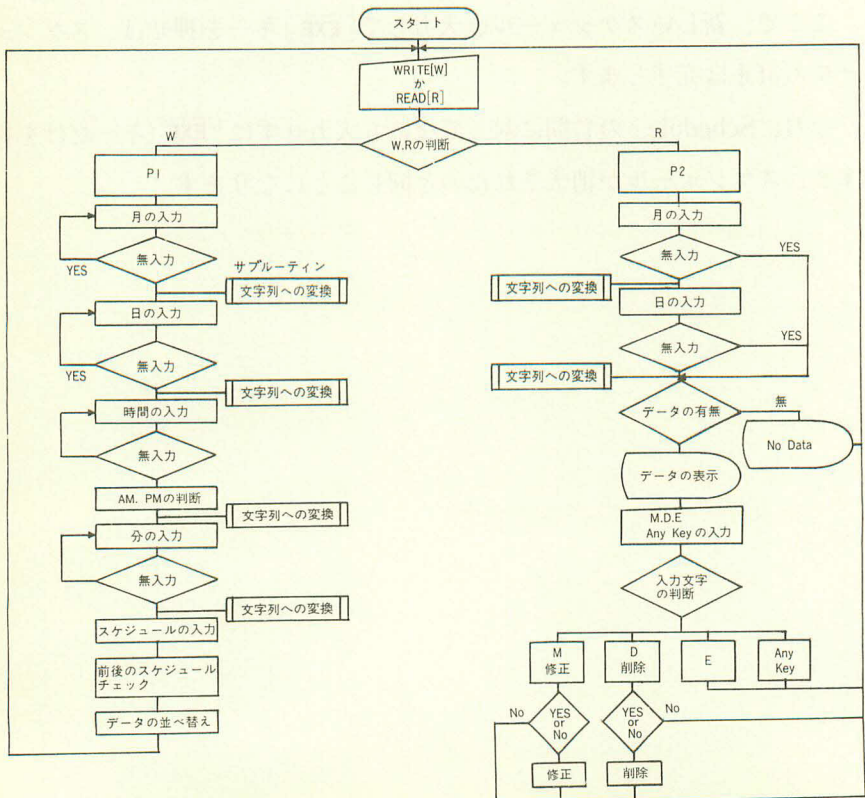
3 スケジュール管理プログラムIII

スケジュール管理プログラムIIIは、さらに使いやすいスケジュール管理プログラムです。プログラムは3部から構成されています。

第1部は、選択のためのメニュー画面のプログラム

第2部は、スケジュール書き込みのためのプログラム

第3部は、スケジュール呼び出しのためのプログラムです。

はじめに、このプログラムの目的と全体の流れを、わかりやすく図解しておきます。

①プログラム入力

まず、WRTモードで SHIFT 0 を指定し、メニュー選択のプログラムを入力します。入力する前に、必ず以前のプログラムをクリアーしてください。

```
P0
  10 PRINT : PRINT "
     Read - [R]";
  20 FOR I=0 TO 100:
      NEXT I
  30 K$= KEY$: IF K$
     ≠"R" THEN IF K$
     ≠"W" THEN 50
  40 GOTO 80
  50 PRINT : PRINT "
     Write - [W]";
  60 FOR I=0 TO 100:
      NEXT I
  70 K$= KEY$: IF K$
     ≠"R" THEN IF K$
     ≠"W" THEN 10
  80 BEEP : PRINT :
     IF K$="R" THEN
     GOSUB #2: GOTO
     100
  90 GOSUB #1
 100 GOTO 10
```

ステップ数：162

次に、**SHIFT** **1** を指定し、スケジュール書き込みのプログラムを入力します。

```
P1
  10 INPUT "Month",$
  20 BEEP : IF $=""
     THEN 10
  30 N= INT  VAL($)
  40 IF N<1 THEN 10
  50 IF N>12 THEN 10
  60 GOSUB 690
  70 A$="I"+$
  80 INPUT "Day",$
  90 BEEP : IF $=""
     THEN 80
 100 N= INT  VAL($)
 110 IF N<1 THEN 80
 120 IF N>31 THEN 80
 130 GOSUB 690
 140 A$=A$+"/"+$
 150 INPUT "Hour",$
 160 BEEP : IF $=""
     THEN 150
 170 N= INT  VAL($)
 180 IF N<0 THEN 150
 190 IF N>23 THEN 15
     0
 200 B$="AM"
 210 IF N≧12 THEN N=
     N-12:B$="PM"
 220 GOSUB 690
 230 B$=B$+$
 240 INPUT "Minute",
     $
 250 BEEP : IF $=""
     THEN 240
 260 N= INT  VAL($)
 270 IF N<0 THEN 240
 280 IF N>59 THEN 24
     0
 290 N=N+100: GOSUB
     690
 300 B$=B$+":"+$
 310 INPUT "Schedule
     ",$
 320 GOSUB 720
 330 RESTORE# : REST
     ORE# "\SCHEDULE
     ",0,350
 340 GOTO 410
 350 RESTORE# "",0,3
     80
 360 READ# $
 370 GOTO 350
 380 WRITE# "\SCHEDU
     LE"
 390 Q=1
 400 GOTO 660
 410 READ# $
 420 P$="\0"
 430 Q$=P$
 440 READ# P$
 450 IF P$="\END.SC"
      THEN 630
 460 READ# H$
 470 IF H$<A$ THEN R
     EAD# $,$: GOTO
     420
 480 READ# I$
 490 IF H$>A$ THEN 5
     10
 500 IF I$<B$ THEN R
     EAD# $: GOTO 42
     0
 510 READ# $
 520 RESTORE# : REST
     ORE# "\SCHEDULE
     ": RESTORE# P$
 530 READ# P$
 540 WRITE# A$,B$,C$
     +D$+E$+F$+G$
 550 A$=H$
 560 B$=I$
 570 GOSUB 720
 580 Q$=P$
 590 READ# P$
 600 IF P$="\END.SC"
      THEN 630
 610 READ# H$,I$
 620 GOTO 510
 630 RESTORE# : REST
     ORE# "\SCHEDULE
     ": RESTORE# P$
 640 $=Q$
 650 Q$= MID$(2):Q=
     VAL(Q$)+1
 660 WRITE# "\"+ STR
     $(Q)+"\,"+A$+",
     "+B$+","+C$+D$+
     E$+F$+G$
 670 WRITE# "\END.SC
     "
 680 RETURN
 690 $=" "+ STR$(N)
 700 $= MID$( LEN($)
     -1)
 710 RETURN
 720 FOR I=0 TO 4
 730 C$(I)=""
 740 NEXT I
 750 I=0
 760 IF LEN($)≦7 THE
     N 800
 770 C$(I)= MID$(1,7
     )
 780 $= MID$(8):I=I+
     1
 790 GOTO 760
 800 C$(I)=$
 810 RETURN
```

ステップ数：931

最後は、 SHIFT 2 を指定し、スケジュール呼び出しのプログラムを入力します。

```
P2
  10 INPUT "Month",$
  20 BEEP : IF $=""
     THEN D$="I": GO
     TO 150
  30 N= INT VAL($)
  40 IF N<1 THEN 10
  50 IF N>12 THEN 10
  60 GOSUB 600
  70 D$="I"+$
  80 INPUT "Day",$
  90 BEEP : IF $=""
     THEN 150
 100 N= INT VAL($)
 110 IF N<1 THEN 80
 120 IF N>31 THEN 80
 130 GOSUB 600
 140 D$=D$+"/"+$
 150 RESTORE# : REST
     ORE# "\SCHEDULE
     ": RESTORE# D$,
     1,420
 160 READ# Y$,$,T$
 170 PRINT : PRINT M
     ID$(2);T$;
 180 READ# $
 190 IF KEY$≠"" THEN
      190
 200 K$= KEY$: IF K$
     ="" THEN 200
 210 BEEP : IF K$="D
     " THEN 450
 220 IF K$="M" THEN
     510
 230 IF K$="E" THEN
     590
 240 L= LEN($)
 250 IF L≦12 THEN PR
     INT : PRINT $;:
      GOTO 350
 260 FOR I=1 TO L-10
 270 IF I=L-10 THEN
     PRINT : PRINT M
     ID$(I);: GOTO 3
     50
 280 PRINT : PRINT M
     ID$(I,11);"→";
 290 IF I=1 THEN IF
     KEY$≠"" THEN 29
     0
 300 K$= KEY$: IF K$
     ="" THEN 300
 310 IF K$≠"D" THEN
     IF K$≠"M" THEN
     IF K$≠"E" THEN
     330
 320 I=100
 330 NEXT I
 340 IF I≧100 THEN 3
     70
 350 IF KEY$≠"" THEN
      350
 360 K$= KEY: IF K$=
     "" THEN 360
 370 BEEP : IF K$="D
     " THEN 450
 380 IF K$="M" THEN
     510
 390 IF K$="E" THEN
     590
 400 RESTORE# D$,1,1
     50
 410 GOTO 160
 420 BEEP : PRINT :
     PRINT "No Data"
     :
 430 IF KEY$="" THEN
      430
 440 BEEP : RETURN
 450 PRINT : PRINT "
     Delete(Y/N)?";
 460 K$= KEY$: IF K$
     ="N" THEN BEEP
     : GOTO 400
 470 IF K$≠"Y" THEN
     460
 480 BEEP : RESTORE#
      : RESTORE# "\S
     CHEDULE": RESTO
     RE# Y$
 490 WRITE#
 500 GOTO 400
 510 PRINT : PRINT "
     Modify(Y/N)?";
 520 K$= KEY$: IF K$
     ="N" THEN BEEP
     : GOTO 400
 530 IF K$≠"Y" THEN
     520
 540 BEEP : RESTORE#
      : RESTORE# "\S
     CHEDULE": RESTO
     RE# Y$
 550 PRINT : INPUT "
     Schedule:",$
 560 READ# Y$,Z$,T$
 570 WRITE# $
 580 GOTO 400
 590 RETURN
 600 $=" "+ STR$(N)
 610 $= MID$( LEN($)
     -1)
 620 RETURN

READY P0
```

ステップ数：855

合計1948ステップ

②プログラムの操作方法

●仕事の選択

RUN モードで [SHIFT] [0] を指定し、メニュー画面を表示します。メニュー画面は、Read- [R] と Write- [W] が交互に表示されます。

書き込みなら [W] を、呼び出しなら [R] を押すと、ビーと鳴って画面が停止し、仕事が選択されます。

●スケジュールの書き込み

画面	入力	キー
Month?	1～12までの数字を入力	EXE
Day?	1～31までの数字を入力	EXE
Hour?	0～23までの数字を入力	EXE
Minute?	0～59までの数字を入力	EXE
Schedule?	30字以内で入力	EXE

ここで再びメニュー画面に戻りますので、書き込みなら再度 [W] を、データの呼び出しなら [R] を押します。

●スケジュールの呼び出し

Month?　　1～12までの数字を入力　[EXE]

＊ [EXE] キーだけを押すと、すべてのデータを呼び出します。

Day?　　1～31までの数字を入力　[EXE]

＊ [EXE] キーだけを押すと、指定した月のデータを呼び出します。

該当データがない場合は、"No Data" が表示されます。

該当データがあれば、月日と時間が表示されます。

スケジュールを見る場合には、[E]、[D]、[M] 以外のキーを押せば、表示されます。

キーを押すたびに、該当データが順次表示されます。

●メニュー画面に戻る

仕事が終了したら終わるという意味で〝END″の[E]キーを押します。

〝No Data″が表示されているときも、[E]キーを押せば、メニュー画面に戻ります。

●データの訂正（スケジュール）

データの訂正を行なうときは、RUNモードでデータを読み出して[M]キーを押します。

〝Modify（Y／N）?″と表示されます。

訂正するときは[Y]キーを押します。

〝Schedule?″と表示されますので、新しいスケジュールを入力し、[EXE]キーを押します。

[N]キーを押せば、訂正されません。

●データの消去（月日、時間、スケジュール）

データを消去するときは、RUNモードでデータを読み出しているときに、[D]キーを押します。

〝Delete（Y／N）?″と表示されます。

消去するときは[Y]キーを押します。

消去しないときは[N]キーを押します。

該当日のデータがなくなると、〝No Data″が表示されます。

以上、2節、3節を通じて、スケジュール管理プログラムの改良を試みてきました。2節のスケジュール管理プログラムIIは、2Kバイトの RAMカードで充分ですが、3節のスケジュール管理プログラムIIIは、4Kバイトの RAMカードを使用しないと、プログラム自体が入りません。

③プログラムの実行

それではプログラムの実行例を見てみましょう。

操作		表示	
[SHIFT] [0]		Read-[R]	交互に繰り返す。書き込みを指定。
		Write-[W]	
[W]		Month?	1～12の数字を入力。
6	[EXE]	Day?	
12	[EXE]	Hour?	
10	[EXE]	Minute?	
15	[EXE]	Schedule?	
EIGYO KAIGI	[EXE]	Read-[R]	交互に繰り返す。読み出しを指定。
		Write-[W]	
[R]		Month?	
6	[EXE]	Day?	
10	[EXE]	6／10 AM11:30	[E], [D], [M] 以外のキーを押して表示を動かします。
[K]		SATO	
[K]		6／10 PM4:00	
[K]		SHOPPING	
[E]		Read-[R]	[E] キーを押すと初期画面に戻ります。
		Write-[W]	
[R]		Month?	
6	[EXE]	Day?	
10	[EXE]	6／10 AM11:30	訂正するときは [M] キーを押します。
[M]		Modify (Y／N)?	Yes, No で答えます。
[Y]		Schedule?	正しいデータを書き込みます。
MEET KOIKE	[EXE]	6／10 AM11:30	訂正が終わりました。
[E]		Read-[R]	[E] キーを押すと初期画面に戻ります。
		Write-[W]	

5

データバンク活用プログラム

カシオデータバンク機能つきポケットコンピュータは、ビジネスにまたホビーに、多方面で愛用されています。

毎日の生活のどんな局面で、どんな働きをしているのか、ここで１人のモニターに登場してもらうことにしましょう。

中村氏は、都心の貿易会社に勤務するビジネスマンです。中堅管理職として、公私ともに忙しい毎日を送っています。仕事にも、友人とのつき合いにも積極的なため、一日の限られた時間のなかで最大限に効率よく行動していくことが、いま彼のテーマになっているようです。

データバンク機能はプログラムなしでも便利に使えますが、プログラムと合わせて使えばさらに活用範囲が広がります。

さっそく、彼の作成したプログラムの一部を公開してもらうことにしました。きっと皆さんの参考になることでしょう。

1 マイル、ヤード換算はお手のもの

外国通にとっても、マイル、ヤードの長さや、ポンド、オンスの重さはメートル法に慣れ親しんでいる日本人には、感覚的に理解しにくいものです。

まして、正確な値を出すとなると、電卓片手にあれこれ計算してみなければなりません。

中村氏も度量衡の換算にはいつも悩まされていました。しかし、データバンクを手に入れてからは、その悩みも解消、度量衡換算の「生き字引」となってしまいました。その秘密は、「度量衡換算プログラム」にあります。

①プログラムの入力

MODE 1 でWRTモードを、SHIFT 0 でプログラムエリアP0を指定します。

```
10 INPUT "Number",
   $
20 IF $="" THEN 10
30 D= VAL($)
40 INPUT "Unit (fr
   om)",$
50 IF $="" THEN 40
60 RESTORE#
70 RESTORE# $+",",
   1,40
80 READ# A
90 INPUT "Unit (to
   )",$
100 IF $="" THEN 90
110 RESTORE#
120 RESTORE# $+",",
   1,90
130 READ# B,$
140 C=D*B/A
150 PRINT C;"[";$;"
   ]"
160 END
```

ステップ数：174

②メモデータの入力

このプログラムを使用するときは、度量衡換算用のデータと他のデータが混在して、計算が正しく行なわれないことを防ぐため、前のメモデータが消去されていることを確認してから、度量衡換算のデータをリストにしたがって入力してください。

メモインモード（MODE 9）を指定して、次の表のデータを入力します。

データ
1, mile, MILE, 1, 60931, km, KM, 1609.31, m, M, 63360, inch, INCH, 5280, ft, FT, 1760, yd, YD, 1, kg, KG, 1000, g, G, 35.2739, oz, OZ, 2.20462, lb, LB,

③プログラムの実行

3500フィートは何メートルになるかを調べてみます。

操　作	表　示	
SHIFT 0	Number?	変換する数を入力
3500 EXE	Unit (from)?	変換する単位からの指定を入力 (Feet)
FT EXE	Unit (to)?	変換したい単位の指定を入力 (Meter)
M EXE	1066. 777462(m)	

④プログラムの応用

度量衡換算プログラムはメモデータに換算の基礎単位を入力しておけば、いろいろな換算ができます。度量衡換算では、ガロンとリッターの関係や尺貫法とメートル法の関係、度量衡換算以外では、為替レートの換算にも利用することができます。

●為替レートをメモデータに追加する

1ドルが240円であれば、ドルをUSD、円をJYEとして入力します。さらに1ポンドが430円であれば、240を430で割った値を、ポンドの表示(STG) とともに入力します。このようにして、フランス・フランはFFR、ドイツ・マルクはDMKの表示で、メモデータに追加すれば、為替レートの変換はお手のものとなります。

追加データ

```
11  1,USD,
12  240,JYE,
13  0.558,STG,
```

⑤参考例

以下のプログラムは、メモデータを利用しないで、すべてのデータをプログラムのなかにおさめた参考例です。このようにデータバンク機能がなければ、変換データをプログラムの中にDATA文として入れなければなりません。

```
P0
  10 INPUT "Number",
     $
  20 IF $="" THEN EN
     D
  30 N= VAL($)
  40 INPUT "Unit (fr
     om)",$
  50 IF $="" THEN 10
  60 A$= MID$(1,7)
  70 INPUT "Unit (to
     )",$
  80 IF $="" THEN 40
  90 B$= MID$(1,7)
 100 RESTORE :J=0
 110 READ $,E
 120 IF $="END" THEN
      160
 130 IF A$=$ THEN C=
     E:J=J+1
 140 IF B$=$ THEN D=
     E:J=J+1
 150 GOTO 110
 160 IF J≠2 THEN PRI
     NT "Not Availab
     le Unit": GOTO
     40
 170 N=N*D/C
 180 PRINT N
 190 GOTO 10
 300 DATA M,1609.31
 310 DATA KM,1.60931
 320 DATA INCH,63360
 330 DATA FT,5280
 340 DATA YD,176092
 350 DATA MIL,1
 360 DATA G,453.592
 370 DATA KG,0.45359
     2
 380 DATA OZ,16
 390 DATA LB,1
 400 DATA END,0
```

ステップ数：378

世界まとめて“What time ?”

国際競争に勝ちぬくためには、情報をいかに早く正確にキャッチするかが、重要なポイントになります。通常の取り引きは、手紙やテレックスで済ますことができます。しかし、一刻を争う状態になると、電話が強力な手段となります。

国際電話をかける場合は、相手の時刻も考慮に入れなければなりません。国際電話をかけるたびに相手先の時刻を調べるようでは、国際ビジネスマンとしてはもう失格です。その点、「世界時計プログラム」は、世界各国の時刻をすべて記憶し、一瞬のうちに教えてくれます。中村氏にとっては、どんなときでも世界中の時間の動きが手にとるようにわかるのです。

①参考例

```
 10 INPUT "Time(hn.
    mm)",$
 20 IF $="" THEN EN
    D
 30 N= VAL($)
 40 A= INTN:B= INT(
     FRACN*100)
 50 IF A≧0 THEN IF
    A≦23 THEN IF B≧
    0 THEN IF B≦59
    THEN 70
 60 GOTO 10
 70 N=A+B/60
 80 INPUT "Place-1"
    ,$
 90 IF $="" THEN 10
100 A$= MID$(1,7)
110 B$= MID$(8,7)
120 INPUT "Place-2"
    ,$
130 IF $="" THEN 80
140 C$= MID$(1,7)
150 D$= MID$(8,7)
160 RESTORE :J=0
170 READ $,G
180 IF $="END" THEN
     220
190 IF A$+B$= MID$(
    1,14) THEN E=G:
    J=J+1
200 IF C$+D$= MID$(
    1,14) THEN F=G:
    J=J+1
210 GOTO 170
220 IF J≠2 THEN PRI
    NT "No Data": G
    OTO 80
230 N=N-E+F:$=""
240 IF N<0 THEN N=N
    +24:$="(B)"
250 IF N≧24 THEN N=
    N-24:$="(A)"
260 A= INTN:B= INT(
     FRACN*60+.5)
270 PRINT A;"h":B;"
    m";$
280 GOTO 10
300 DATA TOKYO,9
310 DATA LONDON,0
320 DATA END,0
```

ステップ数：446

②プログラムの改良

中村氏は、いままでは前記のプログラムを使っていました。しかし、このプログラムでは、世界各地の時刻を行番号300、310のように、プログラムのなかにデータ文として入れなければなりません。

この点、メモデータを利用したプログラムでは、メモインモードでデータを自由に追加することができ、プログラムのなかにデータを書き込むよりも簡単に操作できます。次に、メモデータを利用した中村氏の最新作をみてみましょう。

③プログラムの入力

MODE 1 でWRTモードを、SHIFT 0 でプログラムエリアを指定して、プログラムを入力します。

```
 10 INPUT "Time(hh.
    mm)",$
 20 IF $="" THEN EN
    D
 30 N= VAL($)
 40 A= INTN:B= INT(
     FRACN*100)
 50 IF A≥0 THEN IF
    A≤23 THEN IF B≥
    0 THEN IF B≤59
    THEN 70
 60 GOTO 10
 70 N=A+B/60
 80 INPUT "Place-1"
    ,$
 90 IF $="" THEN 10
100 A$= MID$(1,7)
110 B$= MID$(8,7)
120 INPUT "Place-2"
    ,$
130 IF $="" THEN 80
140 C$= MID$(1,7)
150 D$= MID$(8,7)
160 RESTORE# : REST
    ORE# "",0,280
170 RESTORE# A$+B$,
    0,270
180 READ# $,E
190 RESTORE# : REST
    ORE# C$+D$,0,27
    0
200 READ# $,F
210 N=N-E+F:$=""
220 IF N<0 THEN N=N
    +24:$="(B)"
230 IF N≥24 THEN N=
    N-24:$="(A)"
240 A= INTN:B= INT(
     FRACN*60+.5)
250 PRINT A;"h";B;"
    m";$
260 GOTO 10
270 PRINT "Bad Unit
    ": GOTO 80
280 PRINT "No Data"
    : END
```

ステップ数：400

④データの入力

データの入力はグリニッジ標準時を基準にして、メモインモード（MODE 9）で入力します。世界各地の時刻は、次のリストを参考にしてください。国によっては夏時間が設定されていますので、メモデータを訂正して現状にそくした時刻を入力してください。

データ例

デ　ー　タ
TOKYO, 9
HONG　KONG, 8
BONBAY, 5.5
THEHERAN, 3.5
MOSCOW, 2
ROME, 1
DUSSELDORF, 1
BONN, 1
BERLIN, 1
PARIS, 1
AMSTERDAM, 1
LONDON, 0
NEW　YORK, −5
MIAMI, −5
CHICAGO, −6
LOS　ANGELES, −8
SAN　FRANCISCO, −8
HONOLULU, −10
MADRID, 1
RIO　DE　JANEIRO, −3
BUENOS　AIRES, −3

⑤プログラムの実行

例として、日本時間8時20分がNEW YORKでは何時になるかを調べてみます。

操作		表示	
SHIFT 0		Time (hh. mm)?	調べる時刻を入力します。(時と分の区切りにはピリオドを使います。)
8.20	EXE	Place-1?	東京を入力します
TOKYO	EXE	Place-2?	調べる都市名を入力します。
NEW YORK	EXE	18h 20m (B)	

東京時間　8時20分は、NEW YORKでは前日の18時20分ということがわかります。

(B)は前日を意味しています。また(A)は翌日を意味します。EXEキーを押しますと最初の表示があらわれます。

EXE	Time (hh. mm)?

同様に、HONOLULUの12時はLONDONでは何時かも調べることができます。まず12.00を入力し、Place-1にHONOLULUを、Place-2にLONDONを入力すれば、LONDONの時刻　22h 0mが表示されます。

都市名の入力を間違えますと、"Bad Unit"が表示されますが、EXEキーを押せば、再度Place-1が表示され、時刻はそのままで、都市名の入力だけを行なうことができます。

7 内線番号、市外局番も一発サーチ

社内の内線番号まで自分の電話帳に書き込んでいる人は少ないと思います。しかし、内線番号表から探し出すのも面倒くさいものです。まして、最近はダイヤルインで電話番号と内線番号がいっしょになり、各部ごとに別れていた番号も除々にくずれています。でも中村氏は、「内線番号一発サーチプログラム」で、内線番号を調べなくても、すぐに教えてもらえるようになりました。

①プログラムの入力

MODE 1 、 SHIFT 0 で書き込み用のプログラムを、 SHIFT 1 で呼び出し用のプログラムを入力します。

```
P0
  10 INPUT "1st Data
     ",$
  20 IF $="" THEN EN
     D
  30 FOR Z=0 TO 3
  40 A$(Z)= MID$(Z*7
     +1,7)
  50 NEXT Z
  60 INPUT "2nd Data
     ",$
  70 IF $="" THEN 60
  80 FOR Z=0 TO 3
  90 I$(Z)= MID$(Z*7
     +1,7)
 100 NEXT Z
 110 RESTORE#
 120 RESTORE# ".END"
     ,0,130
 130 WRITE# A$+B$+C$
     +D$+",."+I$+J$+
     K$+L$
 140 WRITE# ".END"
 150 GOTO 10
```

ステップ数：187

```
P1
  10 INPUT "Which Da
     ta",$
  20 IF $="" THEN 10
  30 RESTORE#
  40 RESTORE# $+",",
     0,10
  50 READ# $,$
  60 IF MID$(1,1)≠".
     " THEN READ# $
  70 $= MID$(2)
  80 PRINT $
  90 GOTO 40
```

ステップ数：93

合計280ステップ

②データの入力

データ入力例

部　所	内線番号
SOUMU	4 1 1 1
KEIRI	4 2 1 1
JINJI	4 4 3 1
EIGYOU	4 5 1 1
KOUHOU	4 6 1 1
SENDEN	4 7 1 1

操　作		表　示	
SHIFT 0		1st Data?	部・課名を入力
SOUMU	EXE	2nd Data?	内線番号を入力
4111	EXE	1st Data?	
KEIRI	EXE	2nd Data?	
⋮		⋮	

③プログラムの実行

このように、内線番号を調べたい部・課名を入力すると、内線番号がワンタッチで表示されます。

④プログラムの応用

行番号10を〝City Name〞、行番号60を〝Number〞とし、出力プログラムの行番号10を〝Which City〞に変えれば、今度は立派な市外局番サーチプログラムともなります。

また、内線番号、市外局番をいっしょに記録しておけば、内線番号、市外局番共通のサーチプログラムとなります。

⑤プログラムの改良

内線番号を部・課名だけからでなく、名前からでも調べることができるようにしたのが以下のプログラムです。

WRTモードにしてプログラムエリアP0を指定し、プログラムを訂正してください。

```
10 INPUT "1st Data
   ",$
20 IF $="" THEN EN
   D
30 FOR Z=0 TO 3
40 A$(Z)= MID$(Z*7
   +1,7)
50 NEXT Z
60 INPUT "2nd Data
   ",$
70 IF $="" THEN 60
80 FOR Z=0 TO 3
90 E$(Z)= MID$(Z*7
   +1,7)
100 NEXT Z
```

```
110 INPUT "3rd Data
    ",$
120 IF $="" THEN 11
    0
130 FOR Z=0 TO 3
140 I$(Z)= MID$(Z*7
    +1,7)
150 NEXT Z
160 RESTORE#
170 RESTORE# ".END"
    ,0,180
180 WRITE# A$+B$+C$
    +D$+","+E$+F$+G
    $+H$+","+I$+J$
    +K$+L$
190 WRITE# ".END"
200 GOTO 10
```

ステップ数：263

合計356ステップ

⑥データの入力

データ入力例

部　所	名　　前	内線番号
SOUMU	AKIYAMA	4 1 1 1
SOUMU	MAKI	4 1 1 2
SOUMU	KUROSAWA	4 1 1 3
KEIRI	SATO	4 2 1 1
KEIRI	TAKAHASHI	4 2 1 2
SHOMU	MAKITA	4 3 3 1
JINJI	SUZUKI	4 4 3 1
JINJI	NAGASHIMA	4 4 3 2
JINJI	KIMURA	4 4 3 3

操　作		表　示	
SHIFT 0		1st Data?	部・課名を入力
SOUMU	EXE	2nd Data?	名前を入力
AKIYAMA	EXE	3rd Data?	内線番号を入力
4111	EXE	1st Data?	
⋮		⋮	

⑦プログラムの実行

内線番号を調べるには、Ｐ１のプログラムエリアに入力されているプログラムを使います。

操　作		表　示	
SHIFT 1		Which Data?	MAKI さんの内線番号を調べる
MAKI	EXE	4112	
	EXE	Which Data?	JINJI の内線番号を調べる
JINJI	EXE	4431	
	EXE	Which Data?	

このように、部・課名からでも、名前からでも内線番号を調べることができます。

⑧参考例

前のように改良されたプログラムは、2つの項目からサーチできますから、型番(商品番号)、商品名、価格を入力しておけば、型番からでも、商品名からでも価格を調べることができます。

データ例

```
LIST #
   1 510-01,CITY CYC
     LE,.¥25000
   2 510-02,3-DAN CI
     TY CYCLE,.¥2850
     0
   3 510-03,HAIKIN-D
     AI,.¥9900
   4 510-04,SEITAI-S
     HIATSUKI,.¥6800
   5 510-05,DENSHI-K
     ETSUATSUKEI,.¥7
     500
   6 510-06,KOURAI-N
     INJIN-EKISU,.¥4
     300
   7 510-07,SHOPPING
     -CART,.¥5980
   8 .END
```

4 出張旅費の精算書をプリントアウト

国内、海外を問わず、出張のあとの旅費の精算は面倒くさいものです。出張明けのたまった仕事に忙殺されて、精算はなにかと遅れがちになるものです。そして、いざ旅費の明細ともなれば、1日1日の行動スケジュールと領収書を手がかりにして、おぼろげな記憶にたよらなければなりません。

「出金プログラム」さえあれば万全。その日その日の行動と出金を入力しておけば、毎日の行動スケジュールや支出金額がワンタッチで表示されます。メモデータは、日付順に並び替えられ、その日使った総額も計算されています。

メモデータのプリントアウト一枚が、即、出張費の精算書に早がわりという工合です。

①プログラムの入力

MODE 1 と押してWRTモードを指定し、プログラムエリアP0にはデータ入力のプログラムを、プログラムエリアP1にはデータ出力のプログラムを入力します。

```
P0
 10 INPUT "Date",$
 20 IF $="" THEN EN
    D
 30 N= VAL($)
 40 A= INTN:B= INT(
     FRACN*100)
 50 IF A≥1 THEN IF
    A≤12 THEN 70
 60 GOTO 10
 70 IF B≥1 THEN IF
    B≤31 THEN 90
 80 GOTO 10
 90 $= STR$(100+B)
100 A$= STR$(A)+"."
    + MID$(2)
110 INPUT "Memo-1",
    $
120 IF $="" THEN 11
    0
130 FOR Z=0 TO 3
140 B$(Z)= MID$(Z*7
    +1,7)
150 NEXT Z
160 INPUT "Memo-2",
    $
170 IF $="" THEN 16
    0
180 FOR Z=0 TO 3
190 F$(Z)= MID$(Z*7
    +1,7)
200 NEXT Z
210 INPUT "Kingaku
    ¥",$
220 IF $="" THEN 21
    0
230 Y= INT VAL($)
240 RESTORE#
250 RESTORE# A$,0,5
    10
260 RESTORE# A$+",T
    OTAL,!,"
270 READ# $,$,$,X
280 X=X+Y
290 RESTORE# "",1
300 WRITE# A$,B$+C$
    +D$+E$,F$+G$+H$
    +I$,Y,""
310 FOR Z=0 TO 7
320 B$(Z)=""
330 NEXT Z
340 B$="TOTAL":F$="
    !"
350 READ# J$
360 IF J$=".END" TH
    EN 550
370 READ# $
380 FOR Z=0 TO 3
390 K$(Z)= MID$(Z*7
    +1,7)
400 NEXT Z
410 READ# $,Y
420 RESTORE# "",1
430 WRITE# A$,B$+C$
    +D$+E$,F$+G$+H$
    +I$,X,""
440 A$=J$
450 FOR Z=0 TO 3
460 B$(Z)=K$(Z)
470 F$(Z)= MID$(Z*7
    +1,7)
480 NEXT Z
490 X=Y
500 GOTO 350
510 RESTORE# : REST
    ORE# ".END",0,5
    20
520 WRITE# A$+","+B
    $+C$+D$+E$+","+
    F$+G$+H$+I$+","
    + STR$(Y)+","
530 X=Y
540 GOTO 560
550 RESTORE# : REST
    ORE# ".END"
560 WRITE# A$+",TOT
    AL,!,"+ STR$(X)
    +","
570 WRITE# ".END"
580 GOTO 10
```

ステップ数：742

```
P1
 10 INPUT "Date",$
 20 IF $="" THEN EN
    D
 30 N= VAL($)
 40 A= INTN:B= INT(
     FRACN*100)
 50 IF A≥1 THEN IF
    A≤12 THEN 70
 60 GOTO 10
 70 IF B≥1 THEN IF
    B≤31 THEN 90
 80 GOTO 10
 90 $= STR$(100+B)
100 A$= STR$(A)+"."
    + MID$(2)
110 RESTORE#
120 RESTORE# A$+",T
    OTAL,!,",0,160
130 READ# $,$,$,X
140 PRINT "¥";X
150 GOTO 10
160 PRINT "No Data"
170 GOTO 10
```

ステップ数：198

合計940ステップ

②データの入力

データ入力例

日 付	メ モ 1	メ モ 2	金 額
5．02	SHINKANSEN	TOKYO－OSAKA	12200
5．02	TAXI	TEL	1230
5．02	DINNER／DRINK	MR．ITAKURA	16800
5．03	SHINKANSEN	OSAKA－TOKYO	12200
5．03	TAXI	EKI－JITAKU	2220
5．03	OMIYAGE	BEER	1720

操 作		表 示	
SHIFT 0		Date ?	月日を入力します。日は2桁で入力し、月と日の間には、ピリオド(小数点キー)を入れます。
5.02	EXE	Memo-1?	新幹線を入力
SHIKANSEN	EXE	Memo-2?	東京一大阪を入力
TOKYO-OSAKA	EXE	Kingaku ¥?	金額を入力
12200	EXE	Date ?	最初の質問に戻ります。

Memoの入力は30字以内で入力します。

上記のデータは、Memo-1に新幹線、Memo-2に区間を入力していますが、Memo-1にタクシー、Memo-2に新幹線というように2つの項目を入力し、その合計金額をKingakuに入力するという使い方をしてもかまいません。

同じ月日にデータが多い場合は、数回に分けて入力します。このプログラムは、同じ月日の総合計を自動的に計算してくれます。

③データの追加

5月2日の戸田さんとの昼食代を入力していないことに気がついたとします。このようなデータの追加の場合も、データの入力のときと同じ方法で行ないます。

操作		表示
SHIFT 0		Date?
5.02	EXE	Memo-1?
LUNCH	EXE	Memo-2?
MR.TODA	EXE	Kingaku ¥?
2460	EXE	Date?

結果は、下のリストのように、5月2日の最終行に、LUNCH, MR. TODA, 2460が追加され、同時に5月2日のTOTALの金額が訂正されています。

```
LIST #
  1 5.02,SHINKANSEN        6 5.03,SHINKANSEN
    ,TOKYO-OSAKA,12          ,OSAKA-TOKYO,12
    200,                     200,
  2 5.02,TAXI,TEL,1        7 5.03,TAXI,EKI-J
    230,                     ITAKU,2220,
  3 5.02,DINNER/DRI        8 5.03,OMIYAGE,BE
    NK,MR.ITAKURA,1          ER,1720,
    6800,                  9 5.03,TOTAL,!,16
  4 5.02,LUNCH,MR.T          140,
    ODA,2460,             10 .END
  5 5.02,TOTAL,!,32
    690,
```

④データの出力

データの出力は、RUN モードで SHIFT 1 を指定します。

操　作		表　示	
SHIFT 1		Date?	月日の間にはピリオドを入れ、日は2桁で入力します。
5.02	EXE	¥ 32690	5月2日の総合計の金額が表示されます。
	EXE	Date?	

該当日のデータがない場合は、No Data が表示されます。

さらに EXE キーを押すと、最初の表示に戻ります。

⑤プログラムの応用

出金プログラムは、出張旅費の精算に使うだけでなく、毎日の出金メモ、交通費のメモとして使うこともできます。

交通費のメモとして使う場合は、行番号110の Memo-1 を from に、行番号160の Memo-2 を to に変えれば、「どこからどこまで」の意味となります。

たとえば、from? の質問に SHINJUKU と入力し、to? の質問に TOKYO と入力し、Kingaku ¥?の質問に180と入力すれば、新宿—東京間、180円の交通費を記録することができます。

交通費メモは、タクシーの運転手さんの日計表にも利用でき、FP-12S に接続して使えば、プリントして残すこともできます。

5 売り上げデータは一目りょう然

中村さんは、売り上げデータを日付順にいつも把握できるようにしています。どのお客様からいつ、何を、何個注文を受けたのかをメモデータとして保存し、毎週末にはプリントアウトして保管しています。その結果、日々の営業管理に加え、週単位、月単位のデータの整理がスムースに行なわれています。

①プログラムの入力

MODE 1 と押してWRTモードを指定し、プログラムエリアP0にデータ入力のプログラムを、プログラムエリアP1にデータ出力のプログラムを入力します。

P0

```
 10 INPUT "Date",$
 20 IF $="" THEN EN
    D
 30 N= VAL($)
 40 A= INTN:B= INT(
     FRACN*100)
 50 IF A≧1 THEN IF
    A≦12 THEN 70
 60 GOTO 10
 70 IF B≧1 THEN IF
    B≦31 THEN 90
 80 GOTO 10
 90 $= STR$(B+100)
100 A$= STR$(A)+"."
    + MID$(2)
110 INPUT "Koumoku-
    1",$
120 IF $="" THEN 11
    0
130 FOR Z=0 TO 3
140 B$(Z)= MID$(Z*7
    +1,7)
150 NEXT Z
160 INPUT "Koumoku-
    2",$
170 IF $="" THEN 16
    0
180 FOR Z=0 TO 3
190 F$(Z)= MID$(Z*7
    +1,7)
200 NEXT Z
210 INPUT "Koumoku-
    3",$
220 IF $="" THEN 21
    0
230 FOR Z=0 TO 3
240 J$(Z)= MID$(Z*7
    +1,7)
250 NEXT Z
260 RESTORE#
270 RESTORE# "",0,5
    00
280 READ# Y$
290 IF Y$="!END" TH
     EN RESTORE# : R
     ESTORE# "!END":
     GOTO 500
300 IF VAL(A$)≧ VAL
    (Y$) THEN READ#
     $,$,$,$: GOTO
     270
310 READ# $
320 FOR Z=0 TO 3
330 N$(Z)= MID$(Z*7
    +1,7)
340 NEXT Z
350 READ# $
360 FOR Z=0 TO 3
370 R$(Z)= MID$(Z*7
    +1,7)
380 NEXT Z
390 READ# $
400 RESTORE# "",1
410 WRITE# A$,B$+C$
    +D$+E$,F$+G$+H$
    +I$,J$+K$+L$+M$
420 A$=Y$
430 FOR Z=0 TO 3
440 B$(Z)=N$(Z)
450 F$(Z)=R$(Z)
460 J$(Z)= MID$(Z*7
    +1,7)
470 NEXT Z
480 READ# $
490 GOTO 270
500 $=J$+K$+L$+M$
510 WRITE# A$+","+B
    $+C$+D$+E$+","+
    F$+G$+H$+I$+","
    +$+",!"
520 WRITE# "!END"
530 GOTO 10
```

ステップ数：683

P1

```
10 INPUT "Key Word",
$
20 IF $="" THEN END
30 FOR I=0 TO 3
40 A$(I)= MID$(I*7+1
,7)
50 NEXT I
60 RESTORE#
70 J=0
80 RESTORE# A$+B$+C$
+D$,1,200
90 J=1
100 READ# $
110 PRINT $
120 READ# $
130 PRINT $
140 READ# $
150 PRINT $
160 READ# $
170 PRINT $
180 READ# $
190 GOTO 80
200 IF J=0 THEN PRIN
T "No Data"
210 GOTO 10
```

ステップ数：177

合計860ステップ

②データの入力

次の売り上げデータを入力してみます。

6月21日	東京A支店	XX-CAR	12台
6月21日	大阪B支店	YX-CAR／MK 2	5台
6月21日	横浜C支店	YX-CAR	8台
6月22日	仙台D支店	XX-CAR	6台
6月23日	札幌E支店	YX-CAR	7台
6月24日	博多F支店	XX-CAR	9台
6月25日	広島G支店	XX-CAR	15台
6月28日	四国H支店	XX-CAR	4台

操作		表示	
SHIFT 0		Date?	月と日の間にはピリオドを入れ、日は2桁で入力します。
6.21	EXE	Komoku-1?	支店名を入力
TOKYO-A	EXE	Komoku-2?	商品名を入力
XX-CAR	EXE	Komoku-3?	台数を入力
12	EXE	Date?	3項目の入力が終わると、表示が最初に戻ります。
	⋮	⋮	

データの入力は30字以内で行なってください。データは、日付順に入力しなくても、自動的に日付順に並び替えられます。

③データの出力

入力されたデータは、どの項目からでもサーチすることができます。一例として、XX-CARの売り上げ実績を調べることにします。

以上のように、キーワード「XX-CAR」の該当するデータのすべてを表示します。

同様に、キーワード「OSAKA-B」と入力すれば、大阪B支店の販売データが日付順にすべて表示されます。

該当するデータがなくなると、「Key Word ?」と最初の表示に戻ります。

マトリックス計算

中村氏のところには、さまざまなデータが集まります。そのデータを分析、集計し、一枚の表にまとめあげなければなりません。その際、データの集計表は、項目の数も行も一定ではありません。データごとにタテ・ヨコ計算を一つ一つ行なうのは、たいへんな作業です。

項目と行を指定し、データ入力すれば、タテとヨコの合計を自動的にやってくれるのが、**「マトリックス計算プログラム」**です。

①プログラムの入力

MODE 1 でWRTモードを指定し、SHIFT 0 でプログラムエリアを指定してから、プログラムを入力します。

```
10 INPUT "(m X n)"
   ,A,B
20 A= INTA: IF A≦0
   THEN 10
30 B= INTB: IF B≦0
   THEN 10
40 FOR C=1 TO A
50 FOR D=1 TO B
60 PRINT : PRINT "
   (";C;"x";D;")";
70 INPUT E
80 $="N"+ STR$(D)+
   ","+ STR$(E)
90 IF A=C THEN $=$
   +",0"
100 WRITE# $
110 NEXT D
120 $="M"+ STR$(C)+
   ",0"
130 IF C=A THEN $=$
   +",0"
140 WRITE# $
```

```
150 NEXT C
160 C=B-20: IF C>0
    THEN DEFM C
170 FOR C=1 TO B
180 F(C)=0
190 NEXT C
200 RESTORE#
210 FOR C=1 TO A
220 F=0
230 FOR D=1 TO B
240 READ# $,E
250 F(D)=F(D)+E
260 IF A=C THEN WRI
    TE# F(D)
270 F=F+E
280 NEXT D
290 READ# $: WRITE#
    F
300 NEXT C
310 RESTORE#
320 FOR C=1 TO A
330 RESTORE# "M"
```

```
340 READ# $,E
350 PRINT $;"=";E
360 NEXT C
370 RESTORE#
380 IF A>1 THEN RES
    TORE# "M"+ STR$
    (A-1)
390 READ# $,$
400 F=0
410 FOR C=1 TO B
420 READ# $,E,E
430 F=F+E
440 PRINT $;"=";E
450 NEXT C
460 READ# $,E
470 WRITE# F
480 PRINT "TOTAL=";
    F
490 END
```

ステップ数：470

②プログラムの実行

ここでは、次の表のタテ、ヨコの集計をしてみます。

m / n	商品A	商品B	商品C	商品D	ヨコ計
A支店	5329	4280	3602	2310	
B支店	2682	6313	4203	1128	
C支店	5113	3229	5176	965	
タテ計					総合計

操作		表示	
SHIFT 0		(m×n)?	mは項目数、nは行数 項目数4を入力
4	EXE	?	行数3を入力
3	EXE	(1×1)?	1項目目の1行目を入力
5329	EXE	(1×2)?	1項目目の2行目を入力
2682	EXE	(1×3)?	1項目目の3行目を入力
5113	EXE	(2×1)?	2項目目の1行目を入力
4280	EXE	(2×2)?	
⋮		⋮	

このように、カッコのなかの数字は、左側が項目番号を、右側が行番号を表わしています。最初に指定した行数が終了すると、項目番号も、次の項目番号へと移ります。

最後の数字、4項目目の3行目の965を入力すると、表示は次のように変わります。

操作		表示	
965	EXE	M1＝13124	1項目目のタテの合計
	EXE	M2＝13822	2項目目のタテの合計
	EXE	M3＝12981	3項目目のタテの合計
	EXE	M4＝4403	4項目目のタテの合計
	EXE	N1＝15521	1行目のヨコの合計
	EXE	N2＝14326	2行目のヨコの合計
	EXE	N3＝14483	3行目のヨコの合計
	EXE	TOTAL＝44330	総合計

以上のように、タテ、ヨコの合計および総合計を求めることができます。

例題ではmを項目、nを行としましたが、mを行、nを項目というように、逆に設定することもできます。

データを入力しないで EXE キーを押すと、エラー（ERR 2）が表示されます。AC キーで解除して、入力し直します。

③データの利用

マトリックス計算で入力された数値は、すべてメモデータとして記録されていますので、メモインモードで見ることができます。また、プリンタ（FP-12S）を利用して、次のようなデータをプリントアウトできます。

```
LIST #
  1 N1,5329        7 N3,3229        13 N1,2310,15521
  2 N2,2682        8 M2,13822       14 N2,1128,14326
  3 N3,5113        9 N1,3602        15 N3,965,14483
  4 M1,13124      10 N2,4203        16 M4,4403,44330
  5 N1,4280       11 N3,5176
  6 N2,6313       12 M3,12981
```

なお、マトリックス計算のデータはメモデータとして記録されますので、新しいデータを入力するときは、WRTモードでNEW #を入力してメモデータを消去してからプログラムを実行してください。

7 モールス信号はヨットに不可欠

中村氏の趣味は、学生時代におぼえたヨットです。現在は、友人たちとの共有で小型のクルーザーを持っています。

無線電話が発達したといっても、トン・ツー・トン・ツーのモールス信号がなくなったわけではありません。時折、机の上の人指し指が、リズミカルな動きでモールス信号をたたいています。

①プログラムの入力

[MODE] [1] と押してWRTモードを、[SHIFT] [0] と押してプログラムエリアを指定します。

```
 10 REM * MORSE
 20 RESTORE# "MORSE
     CODE",0,900
 30 A$= KEY$
 40 IF A$="" THEN 3
    0
 50 RESTORE# : REST
    ORE# "MORSE COD
    E": READ# $
 60 RESTORE# A$,1,3
    0
 70 READ# $,B$
 80 IF B$≠A$ THEN 3
    0
 90 PRINT : PRINT A
    $;" ";$;
100 FOR I=1 TO LEN(
    $)
110 IF MID$(I,1)="-
    " THEN GOSUB 20
    0: GOTO 130
120 GOSUB 300
130 NEXT I
140 GOTO 30
200 BEEP 1: BEEP 1:
     BEEP 1
210 RETURN
300 BEEP 1
310 RETURN
900 PRINT "NO DATA"
910 END
```

ステップ数：220

②データの入力

WRT モードで NEW #を入力し、メモデータを消去してから、以下のモールス信号を入力します。

データの入力は、メモインモード（MODE 9）で行ないます。

センターピリオドは、拡張モード（MODE ・）のシフトインモードで X キーを押します。

入力データ

MORSE CODE	——, M	——・・, Z
・—, A	—・, N	・————, 1
—・・・, B	———, O	・・———, 2
—・—・, C	・——・, P	・・・——, 3
—・・, D	——・—, Q	・・・・—, 4
・, E	・—・, R	・・・・・, 5
・・—・, F	・・・, S	—・・・・, 6
——・, G	—, T	——・・・, 7
・・・・, H	・・—, U	———・・, 8
・・, I	・・・—, V	————・, 9
・———, J	・——, W	—————, 0
—・—, K	—・・—, X	
・—・・, L	—・——, Y	

③プログラムの実行

アルファベットや数字を入力すると、モールス信号が表示され、同時にモールス信号のように鳴ります。MODE 0 で RUN モードを指定してください。

操 作	表 示
SHIFT 0	
C	C —・—・
A	A ・—
S	S ・・・
I	I ・・
O	O ———

小遣い管理帳

給料日が近くなると、だれでも気になるのが財布の中身。中村氏にとっても赤字を出さないためのやりくりはたいへんなものです。特にクレジットカードでの支払いは、ついつい忘れてしまいがちです。しかし、「小遣い管理帳」が登場してからは、そんな心配は一掃。銀行口座を含めて、小遣い管理はパーフェクトです。

①プログラムの入力

MODE 1 と押して WRT モードを、SHIFT 0 と押してプログラムエリアを指定します。

```
 10 INPUT "Date",$
 20 IF $="" THEN 24
    0
 30 D$= MID$(1,7)
 40 INPUT "Object",
    $
 50 IF $="" THEN 40
 60 FOR B=0 TO 4
 70 E$(B)= MID$(1,7
    )
 80 $= MID$(8)
 90 NEXT B
100 INPUT "Kingaku"
    ,$
110 IF $="" THEN 10
    0
120 N= VAL($)
130 IF E$≠"NYUKIN"
    THEN N=-N
140 RESTORE# : REST
    ORE# "\Zan",,19
    0
150 READ# A$,S
160 S=S+N
170 RESTORE# : REST
    ORE# "\Zan"
180 GOTO 200
190 S=N
200 WRITE# D$,E$+F$
    +G$+H$+I$+","+
    STR$(N)
210 WRITE# "\Zan,"+
     STR$(S)
220 PRINT "Zandaka
    ¥";S
230 GOTO 10
240 S=0
250 RESTORE# : REST
    ORE# "\Zan",,27
    0
260 READ# A$,S
270 PRINT "Zandaka
    ¥";S
280 GOTO 10
```

ステップ数：338

②プログラムの実行

さっそく、RUNモードで記帳してみましょう。ここでは、下記のようなデータを入力してみました。

日　付	内　　容	金　額
6／21	NYUKIN	48000
6／22	BOOK	2800
6／22	COFFEE	900
6／23	DRINK WITH ARAI	13000
6／26	Y－SHIRTS	4800
6／30	NYUKIN	10000

操　作		表　示	
SHIFT　0		Date?	月日を入力します。どのような形式で入力してもかまいません。
6／21	EXE	Object?	使用内容を入力します。入金の場合はNYUKINと入力します。
NYUKIN	EXE	Kingaku?	金額を入力します。
48000	EXE	Zandaka ¥48000	現在の残高が表示されます。
	EXE	Date	最初の質問に戻ります。
6／22	EXE	Object?	
BOOK	EXE	Kingaku?	
2800	EXE	Zandaka ¥45200	
⋮	⋮	⋮	

以上のように、そのときどきの残高が表示されます。

「NYUKIN」の入力以外は、すべて支出として扱われます。

また、支出が残高を超えた場合は、－(マイナス)で残高が表示されます。

●残高の確認

現在の残高だけを確認したい場合は、Date？の表示に続けて EXE キーを押します。

ポケットコンピュータ　PB-410/FX-720P
データバンク活用ハンドブック

昭和59年7月1日　改訂1刷発行

著　者　山　田　靖　二
編集者　株式会社　新晃社
発行所　カシオ計算機株式会社
東京都新宿区西新宿2-6-1
〒160-91 新宿住友ビル内
電話 (03)347-4811(大代表)
印刷所　株式会社　新晃社